# MSX

## truuks en tips

### deel 1

A.C.J. Groeneveld

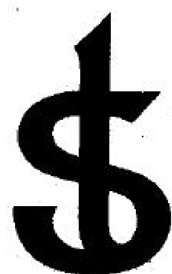


# MSX

## truuks en tips

## deel 1

A.C.J. Groeneveld



uitgeverij STARK - TEXEL
postbus 302 - 1794 ZG Oosterend tel. 02223 - 661

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Groeneveld, A.C.J.

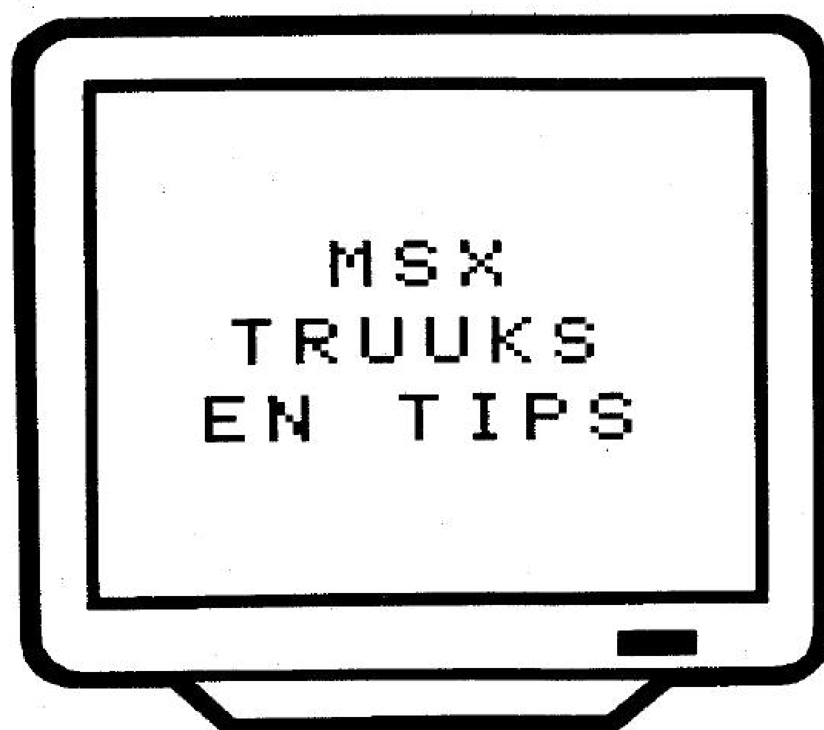
MSX truuks en tips / A.C.J. Groeneveld.—Oosterend: Stark-Texel
CDI. 1J
ISBN 90 6398 900 8
SISO 365.3 UDC 681.3.06
Trefw.: programmeren (computer)/MSX (computer).

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

1e druk 1985
ISBN 90 6398 900 8

**INHOUD** **pag.**

# Om te beginnen



Eindelijk is het dan zover. De MSX-standaard is doorgebroken. Na veel toestanden is er dan eindelijk een standaard computer op de hobbymarkt verkrijgbaar. Dat werd eens tijd.

Ook u hebt het verstandige besluit genomen om een MSX computer aan te schaffen. Welk merk u ook hebt gekozen, het MSX-logo staat voor veel-computer-voor-uw-geld.

In dit boekje is een keur van MSX truuks en tips opgenomen. Niet zo maar wat kleine grapjes maar korte, krachtige stukjes programmatuur waar u bij het programmeren heel veel aan kunt hebben.

Dat uw keuze voor een MSX-computer de juiste was, zult u al bladerend in dit boekje al heel snel beseffen. Het ongeëvenaarde MSX-basic maakt het mogelijk om kort en heel krachtig te programmeren. Programmeertruuks die op een andere computer vaak bladzijden programmeerwerk vergen, kunt u in MSX-basic in enkele regels voor elkaar krijgen.

Zo vindt u in dit eerste deel van de serie MSX truuks en tips voornamelijk korte maar verdraaid krachtige routines.

Eerst even drie programma's om de smaak te pakken te krijgen...

Met het eerste programma bent u voor eens en voor altijd van uw cassetterecorder twijfels af. Het wekker-programma waarschuwt u wanneer u de tijd vergeet omdat u zo ver in uw computerhobby opgaat. Eventueel kan de cassetteratel dienen als wekgeluid...

Succes.

# Cassette checker

MSX is op cassette helaas (nog) niet zo sterk! Vaak kan het gebeuren dat een op cassette geplaatst programma niet korrekt wordt geladen. MSX geeft daar echter geen melding van !!!

Het volgende programma van drie regels kan boven elk programma worden geplaatst en stelt wanneer u het programma uitvoert, op regel 1 automatisch een kontrolegetal samen. Staat op regel 1 echter al een kontrolegetal, dan kontroleert dit programma of het kontrolegetal wel klopt. Een eventuele fout wordt onmiddellijk gemeld.

U kunt dit programma het beste als volgt gebruiken:

U wilt een programma op cassette SAVEn
1: geef een RENUM 65529,65529 in
2: zorg dat op regel 1 het kontrolegetal gelijk is aan XXXXXXXX (8 maal een X)
3: RUN het programma een keer, onderbreek het en kontroleer of op regel 1 een kontroleergetal werd samengesteld
4: SAVE het programma nu

U wilt het programma LOADen, kontroleren en gebruiken
laad het programma en RUN het. Wanneer het programma geen "CHECKFOUT IN PROGRAMMA" geeft, kunt u er voor bijna 100% van overtuigd zijn dat het programma ook werkelijk goed geladen is en geen fouten vertoont die aan een LOAD zijn te wijten.

De programmalijst:

```
1 REM XXXXXXXX = CONTROLEGETAL
2 C=0:FOR I=32784! TO PEEK(63170!)+256*P
EEK(63171!)-1:C=C+PEEK(I):NEXT I:C$=RIGH
T$(STR$(C+100000000#),8):IF PEEK(32775!)
=88 THEN FOR I=1 TO 8:POKE 32774!+I,ASC(
MID$(C$,I)):NEXT I
3 FOR I=1 TO 8:IF CHR$(PEEK(32774!+I))<>
MID$(C$,I,1) THEN PRINT "CHECKFOUT IN PR
OGRAMMA":STOP ELSE NEXT I
4 C$="":C=0:I=0
```

# Wekker

De computer kan een verslavend ding zijn. Voor je het weet, zit je weer tot diep in de nacht mooie figuren te tekenen of prachtige klankenreeksen te produceren.

Het volgende programma geeft u de mogelijkheid om in een programma een wekker in te bouwen. Tussen regel 20 en 9999 kunt u elk mogelijk programma opnemen. Op de bovenste regel vraagt de computer u bij uitvoering over hoeveel minuten u een waarschuwing wenst. Vervolgens wordt bovenin het beeld de tijd naar nul afgeteld waarna een geluidssignaal wordt gegeven en u opnieuw een aantal minuten kunt ingeven.

Het grappige is, dat (behalve op regel 1) uw beeldscherm-indeling niet wordt aangetast door deze routine.

Om deze routine 'kaal' uit te proberen, kunt u op regel 20 het beste even een:

```
20 GOTO 20
```

opnemen.

Pas op: tijdens floppy-bevelen, cassettebevelen of tijdens een (LINE) INPUT kommando wordt de tijd niet bijgehouden en loopt u vertraging op! Wanneer u de elders in dit boek vermelde ingaveroutine gebruikt, loopt de tijd gewoon door tijdens ingave.

De programmalijst:

```
1 REM ****************************
2 REM *          WEKKER          *
3 REM *          ------          *
4 REM ****************************
6 REM
10 CLS:SCREEN 0:WIDTH 40:KEY OFF:GOSUB 1
0000:ON INTERVAL=50 GOSUB 10000:INTERVAL
 ON
10000 CX=POS(0):CY=CSRLIN:IF TI=0 THEN L
```

```
OCATE 0,0:INPUT "OVER HOEVEEL MINUTEN WA
ARSCHUWEN ";TI:LOCATE 0,0:PRINT SPACE$(4
0);:LOCATE CX,CY:TI=INT(TI*60):RETURN
10010 TI=TI-1:IF TI=0 THEN PLAY "M9999L8
T128S1CDEFGFEDCDEFGFEDC":LOCATE 0,0:PRIN
T "HET IS TIJD !!!";SPACE$(40-POS(0)); E
LSE LOCATE 0,0:PRINT USING "& ## & ## &"
;"NOG";TI\60;"MINUTEN EN";TI-60*(TI\60);
"SECONDEN";:LOCATE CX,CY:RETURN
10020 IF PLAY(0) THEN 10020 ELSE RETURN
```

# Cassettegeratel

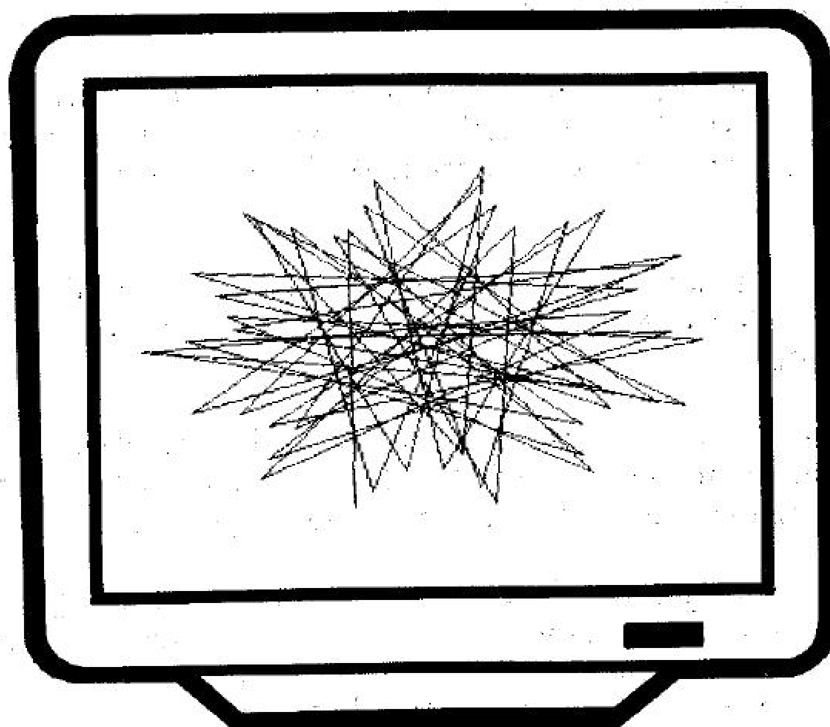
Het volgende programma maakt gebruik van de aansturing van de cassetterecorder om een ratelend geluid te maken. Misschien leuk om bij een spelletje te gebruiken.

In dit geval houdt het geratel bij een toetsaanslag op. Door de FOR-NEXT loop te veranderen, kan het geratel sneller of langzamer worden gemaakt.

De programmalijst:

```
10 REM *************************
20 REM *   CASSETTEMOTORRATEL  *
30 REM *   ------------------  *
40 REM *************************
50 REM
60 MOTOR:FOR I=1 TO 10:NEXT I:IF INKEY$=
"" THEN 60 ELSE MOTOR OFF:STOP
```

# Muziek maken op de MSX



Wie MSX zegt, zegt MUZIEK. Geen wonder; nog nooit was het programmeren van muziek op een computer zo gemakkelijk.

Echter, zonder toepassing van een paar kleine truukjes komen we er met muziek maken vaak toch niet helemaal uit.

Op de volgende bladzijden beginnen we met een elektronisch-orgelprogramma. Misschien een beetje groot woord maar het effekt van dit hele korte programma is toch erg leuk. Bovendien is het programma gemakkelijk te veranderen.

Verderop gaan we wat verder op muziek als verschijnsel in. Het glissando-orgeltje doet een beetje synthersizer-achtig aan en geeft een heel leuk effekt.

De drieklanker vormt zowel voor de muziek-theoreticus als voor de enigszins muzikale leek een bijzonder leuk speelgoed waarmee vele liedjes met een enkele toets kunnen worden begeleid.

In het laatste programma kan het gedrag van een samengestelde toon worden bestudeerd. Bijzonder interessant voor de terzakenkundige maar ook voor de leek zondermeer de moeite waard, al was het maar voor de mooie figuren die geproduceerd kunnen worden...

# Orgeltje

Het volgende programmaatje maakt van uw MSX-computer een klein electronisch orgeltje.

De letter -C- stelt ook daadwerkelijk de C voor. De letters S, F, G, J, K, L en de aanhalingsteken-toets vormen de zogenaamde 'zwarte' toetsen.

'Vader Jacob' moet iedereen lukken met dit programma...

De progammalijst:

```
10 REM *************************
20 REM *       ORGELTJE        *
30 REM *       --------        *
40 REM *************************
50 REM
60 POKE 64683!,255:DEFUSR=144:KEY OFF:CL
S:V$="AZSXCFVGBNJMK,L./'"
70 K$=INKEY$:IF K$="" THEN 70
80 K=INSTR(V$,K$):IF K=0 THEN 70
90 K=31+K:U=USR(0):PLAY"T32L1M9999S1N=K;
":GOTO 70
```

# Glissando

Dit programma lijkt wel wat op het eerdere orgel-programma. Nu worden echter onmiddellijk de SOUND registers aangesproken. Het programma heeft als speciaal effekt dat wanneer een volgende toon wordt gespeeld terwijl de vorige nog niet is uitgeklonken, er een glissando wordt gespeeld; de oude toon "glijdt" naar de nieuwe toe. Een heel leuk effekt, probeert u maar...

De programmalijst:

```
10 REM ***************************
20 REM *          GLISSANDO          *
30 REM *          ---------          *
40 REM ***************************
50 REM
60 POKE 64683!,255:DEFUSR=144:KEY OFF:CL
S:V$="AZSXCFVGBNJMK,L./'"
70 DIM F(LEN(V$)):FOR I=0 TO LEN(V$):F(I
)=2^(I/12):NEXT I
80 F=F-(F-F1)/2:GOSUB 120:V=-(V-1)*(V>0)
:F=-F*(V>0):SOUND 8,V:K$=INKEY$:IF K$=""
 THEN 80
90 K=INSTR(V$,K$):IF K=0 THEN 80
100 F1=111760!/(400*F(K)):V=15:IF F=0 TH
EN F=F1
110 GOTO 80
120 SOUND 0,F-256*INT(F/256):SOUND 1,INT
(F/256):SOUND 0,F-256*INT(F/256):RETURN
130 SOUND 1,INT(F/256):SOUND 0,F-256*INT
(F/256):RETURN
```

# Drieklanker

Het volgende programma is zowel voor de muziek-theoreticus als voor de muzikale leek een heel aardig stuk speelgoed.

Op de onderste rij toetsen is de kwintencirkel geprojecteerd. De C zit ook daadwerkelijk onder de letter C. Rechts zitten de mollen (F, BES, ES, AS, DES, GES,...) en links de kruizen (G, D, A, E, B, FIS, CIS,...). Aanraking van een van de toetsen resulteert in een grote drieklank met als grondtoon de toon uit de kwintencirkel. De bovenliggende toets beschikt op dezelfde wijze over de kleine drieklank. Daarboven ligt weer het septime akkoord (zonder kwint).

De pijl omhoog resulteert hierin, dat de eerstvolgende grote drieklank een overmatige terts krijgt. De pijl omlaag zorgt op eenzelfde wijze voor een overmatige kwint.

De drieklanken zijn door middel van hun omdraaiingen steeds binnen een vast octaaf gehouden, gerelateerd aan een vaste grondtoon. Deze vaste grondtoon kan worden verschoven met behulp van de bovenste toetsen. De standaardinstelling komt overeen met toets -5-. Het intoetsen van een 4 heeft zo dus tot gevolg dat de eerstvolgende drieklank een halve toon lager wordt uitgevoerd. Een 7 heeft een verhoging van een gehele toon tot gevolg.

Dit verhaal zal de leek doen duizelen. Echter, wanneer het programma eenmaal is ingetoetst, zal blijken dat de muzikale computeramateur met behulp van drie of meer toetsen al snel allerlei liedjes kan begeleiden. Voor het meest eenvoudige werk zijn drie op de onderste balk naast elkaar liggende toetsen vaak al voldoende om een 'volledige' begeleiding te verzorgen.

Een kort programma waar u vele uren zoet mee zult zijn...

De programmalijst:

```
10 REM **************************
20 REM *      DRIEKLANKER       *
30 REM *      -----------       *
40 REM **************************
```

```
50 REM
60 POKE 64683!,255:DEFUSR=144:KEY OFF:CL
S:SCREEN ,,0:V$="/;P.LO,KIMJUNHYBGTVFRCD
EXSWZAQ":W$="1234567890-=\":L$="":G=45
70 K$=INKEY$:IF K$="" THEN 70 ELSE IF AS
C(K$)=30 THEN T4=(T4=0):T5=0:GOTO 70 ELS
E IF ASC(K$)=31 THEN T5=(T5=0):T4=0:GOTO
 70
80 K=INSTR(V$,K$):IF K=0 THEN K=INSTR(W$
,K$):IF K=0 THEN U=USR(0):GOTO 70 ELSE G
=40+K:K=L:GOTO 70
90 IF K=0 THEN 70 ELSE T1=-2*(K MOD 3>0)
+INT((K-1)/3)*7:T2=T1+2-(2-T4)*((K-1)MOD
 3=0)-1*((K-2)MOD 3=0):T3=T1+6-(K MOD 3>
0)+T5*((K-1)MOD 3=0):T1=T1 MOD 12+G:T2=T
2 MOD 12+G:T3=T3 MOD 12+G:T4=0:T5=0
100 U=USR(0):PLAY "M9999S1T32L1N=T1;","M
9999S1T32L1N=T2;","M9999S1T32L1N=T3;":GO
TO 70
```

# Golfpatronen

Zoals u weet, bestaat geluid uit trillingen. Deze trillingen zijn voor te stellen als golvende lijnen. Wanneer we meerdere tonen tegelijk waarnemen, nemen we eigenlijk slechts een enkele trilling waar. Deze trilling kan echter nogal complex van vorm zijn.

Het volgende programma vraagt van u de ingave van drie frequenties. Maximaal drie keer kunt u het aantal trillingen per seconde ingeven van de toon die u wilt horen. De computer laat u vervolgens deze tonen door elkaar horen en tekent daarbij twee regelmatige eenheden van de resulterende trilling.

Voor leken is het grappig om te zien hoe de toon er uit ziet die u hoort. Geef voor het betere resultaat steeds getallen tussen 200-5000 in.

Voor diegene die er iets meer van afweet kan dit programma een interessant stukje studiemateriaal zijn.

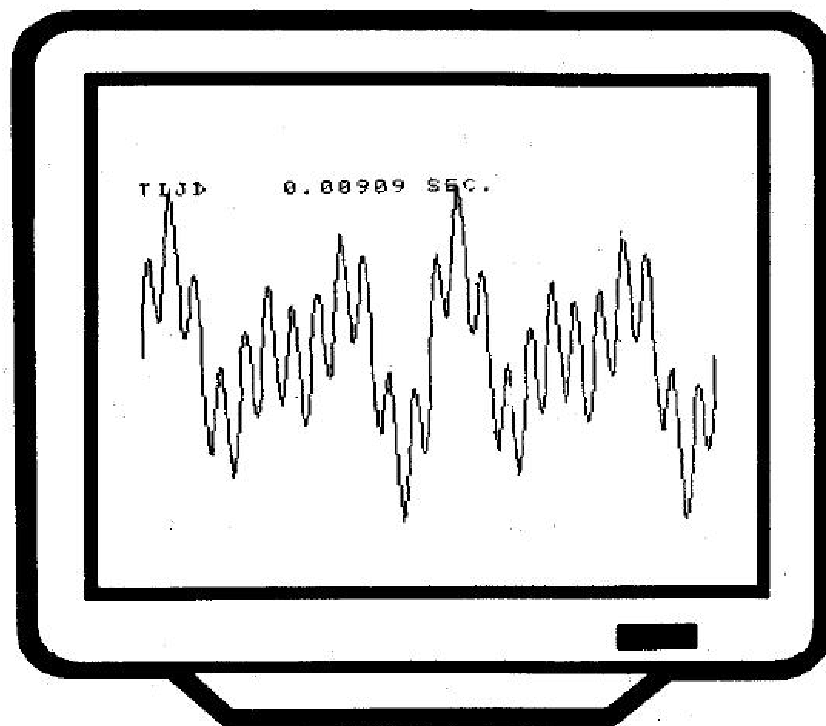
Probeert u bijvoorbeeld maar eens een toon van 400 Hz en een toon van 420 Hz door elkaar te laten klinken en kijk hoe mooi het programma de bij deze kombinatie optredende zweving in beeld brengt...

De programmalijst:

```
10 REM ***************************
20 REM *      GOLFPATRONEN        *
30 REM *      ------------        *
40 REM ***************************
50 REM
60 BEEP:SCREEN 0:CLS
70 DIM HZ(2),PO(2,1)
80 FOR I=0 TO 2:PRINT USING "& #&";"AANT
AL HERZ";I+1;"e STEM ";:INPUT HZ(I):NEXT
 I
90 FOR I=0 TO 2:IF HZ(I)>0 THEN PO=11176
0!/HZ(I):PO(I,1)=INT(PO/256):PO(I,0)=PO-
256*PO(I,1)
100 NEXT I
```

```
110 FOR I=0 TO 2:IF HZ(I)>0 THEN SOUND 2
*I,PO(I,0):SOUND 2*I+1,PO(I,1)
120 NEXT I
130 FOR I=8 TO 10:IF HZ(I-8)>0 THEN SOUN
D I,15
140 NEXT I
150 T=2:Q=2:GOSUB 260:IF M THEN GOTO 190
160 Q=3:GOSUB 260:IF M THEN GOTO 190
170 FOR I=3 TO 1E+20 STEP 2:FOR J=3 TO S
QR(I) STEP 2
180 IF I/J=INT(I/J) THEN NEXT I ELSE NEX
T J:Q=I:GOSUB 260:IF M=0 THEN NEXT I
190 Q=0:F=0:FOR I=0 TO 2:IF HZ(I)>0 THEN
 F=F+1
200 NEXT I:SCREEN 2
210 M=0:FOR I=0 TO 12.57 STEP .04:Y=96-(
SIN(HZ(0)*I)+SIN(HZ(1)*I)+SIN(HZ(2)*I))*
96/F:X=20.37*I
220 IF M=0 THEN PSET(X,Y):M=-1 ELSE LINE
 -(X,Y)
230 NEXT I
240 OPEN "GRP:" AS 1:PSET (0,0):PRINT #1
,USING "& ###.##### &";"TIJD ";T;"SEC.":
CLOSE
250 IF INKEY$="" THEN 250 ELSE RUN
260 M=0:FOR K=0 TO 2:IF HZ(K)/Q=INT(HZ(K
)/Q) THEN NEXT K:FOR K=0 TO 2:HZ(K)=HZ(K
)/Q:NEXT K:T=T/Q:GOTO 260
270 FOR K=0 TO 2:IF HZ(K)>0 AND HZ(K)<Q
THEN M=-1:RETURN ELSE NEXT K:RETURN
```

# Serieus en minder serieus tekenwerk



Iedere MSX-computerbezitter weet dat zijn of haar computer in een handomdraai het mooiste tekenwerk kan leveren.

Het maken van tekeningen kan natuurlijk een plezierig tijdverdrijf zijn. Maar ook voor het serieuze tekenwerk (grafieken, funkties) deinst uw MSX-computer niet terug.

Op de volgende bladzijden beginnen we met het haken van een kleedje dat u jammer genoeg nooit van het beeldscherm af zult kunnen halen... Een heel fraai effekt in een paar programmaregels.

Op de daaropvolgende bladzijden volgt steeds iets serieuzer tekenwerk. Vooral het spirograph-programma is zeker de moeite van het intikken waard. In een tiental programmaregels speelt u dit spel waarbij de mooiste spiraalfiguren ontstaan.

Het arceren (instrepen, schuin of recht) van rechthoeken en cirkels is voor vele aspirant-programmeurs een ondoenlijke opgave. U zult echter zien dat het karwei in een vijftal goed gekozen programmaregels geklaard kan worden.

Het lissajous-programma maakt voor u leerzame en fraaie patronen op het beeldscherm.

Uiteindelijk is een "speciaaltje" opgenomen voor de middelbare scholier. Geen geknoei op rasterpapier; laat de MSX-computer maar uitzoeken hoe die lastige funktie loopt...

# Kleedje haken

Het volgende programma biedt een leuk grafisch grapje, misschien te gebruiken als intro voor een spelprogramma of iets dergelijks.

Door een aantal punten op een cirkel te plaatsen en vervolgens alle punten met een lijn met elkaar te verbinden, ontstaat een grappig patroon dat wel wat op een gehaakt kleedje lijkt...

De programmalijst:

```
10 REM **************************
20 REM *      KLEEDJE HAKEN     *
30 REM *      -------------     *
40 REM **************************
50 REM
60 SCREEN 2:K=15
70 DIM A(K,1):FOR I=0 TO K
80 A(I,0)=COS(I/K*6.28)*125+125
90 A(I,1)=SIN(I/K*6.28)*90+90
100 NEXT I
110 FOR I=0 TO K-1:FOR J=I+1 TO K
120 LINE (A(I,0),A(I,1))-(A(J,0),A(J,1))
130 NEXT J:NEXT I
140 GOTO 140
```

# Random tekening

Het volgende programma is bedoeld als een interessant 'speelgoedje'.

Door twee willekeurige sinusoiden met lijnen met elkaar te verbinden genereert dit programma steeds nieuwe patronen van lijnen.

Meestal zijn deze patronen het aanzien niet waard maar een enkele keer ontstaat toevallig plotseling een heel mooi lijnenspel.

Bevalt een tekening u tijdens het ontstaan al niet? Geef maar een toets in. De computer maakt onmiddellijk een andere tekening voor u.

De programmalijst:

```
10 REM *************************
20 REM *    RANDOM TEKENING    *
30 REM *    ---------------    *
40 REM *************************
50 REM
60 SCREEN 2:F1=RND(-TIME):F1=RND(1):F2=R
ND(1):F3=RND(1):V1=RND(1):V2=RND(1)
70 FOR I=0 TO 12.6 STEP .02+RND(1)
80 X1=20*I:Y1=(SIN(I*2*F1)+1)*92*V1
90 X2=(COS(I*2*F2+F3*6.3)+1)*128*V2:Y2=1
6*I
100 LINE (X1,Y1)-(X2,Y2)
110 IF INKEY$="" THEN NEXT I ELSE RUN
120 IF INKEY$="" THEN 120 ELSE RUN
```

# Spirograph

Wie kent het spel SPIROGRAPH niet? Het spel waarmee je de mooiste figuren kunt tekenen door twee tandwielen langs elkaar te laten lopen. Het ene tandwiel zet je vast en het andere draai je daar omheen. Door een pen in een gaatje van dit tweede tandwiel te steken, maak je tijdens het ronddraaien de mooiste, spiraalachtige figuren.

Het volgende programma simuleert dit prachtige spel. Het programma vraagt u allereerst om de stralen van de te gebruiken wieltjes. Vervolgens vraagt het programma u, op welke afstand van het middelpunt van wiel 2 de pen in het gaatje dient te worden gestoken. Hierna wordt de resulterende figuur op beeldscherm getekend.

Het grappige is, dat u met dit programma niet gebonden bent aan de beperkingen die het gebruik van echte tandwielen oplegt. U kunt bijvoorbeeld de stralen ook negatief ingeven en een pen-positie kiezen die buiten het tweede wiel ligt.

Wanneer u de tweede straal negatief ingeeft, dan krijgt u het beeld dat ontstaat wanneer het tweede wiel zich BINNEN het eerste wiel ronddraait. Hieruit komen meestal de mooiste figuren.

Het programma blijft net zolang tekenen totdat u een toets ingeeft. Na deze ingave kunt u weer nieuwe gegevens ingeven.

Kunt u dit verhaal niet helemaal volgen? Gewoon intikken, even nakijken en dan runnen. Geef maar wat rare waarden in, tien tegen één dat u er op een gegeven moment een heel mooi figuur uit krijgt!

De programmalijst:

```
10 REM ***************************
20 REM *       SPIROGRAPH        *
30 REM *       ----------        *
40 REM ***************************
50 REM
60 SCREEN 0:WIDTH 40:KEY OFF:CLS
70 INPUT "STRAAL VAN WIEL 1  ";R1
80 INPUT "STRAAL VAN WIEL 2  ";R2
```

```
90 INPUT "AFSTAND TEKENPUNT  ";R3
100 SCREEN 2
110 FX=128/(R1+R2+R3):FY=96/(R1+R2+R3)
120 FOR T=0 TO 1E+20 STEP .4
130 X=FX*(COS(T)*R3+COS(T*R2/R1)*(R1+R2)
)+128
140 Y=FY*(SIN(T)*R3+SIN(T*R2/R1)*(R1+R2)
)+96
150 IF M=0 THEN PSET (X,Y):M=1 ELSE LINE
 -(X,Y)
160 IF INKEY$="" THEN NEXT T ELSE RUN
```

# Arceer rechthoek horizontaal

Het arceren van op beeldscherm getekende figuren is heel lastig te programmeren. Het bedenken van een juiste arceerroutine kost de meeste amateurs enorme hoofdbrekens.

Toch kan een figuur betrekkelijk eenvoudig worden gearceerd wanneer we wat slimmigheden toepassen.

De volgende subroutine dient 'gevoed' te worden met vijf variabelen:

| | |
|---|---|
| X1,Y1 | eerste hoofdpunt rechthoek |
| X2,Y2 | tweede hoofdpunt rechthoek |
| D | afstand tussen twee arceerstrepen |

Een GOSUB 1000 doet het verdere werk; de gearceerde rechthoek wordt getekend.

De programmalijst:

```
1000 REM ***************************
1010 REM * ARCEER RECHTHOEK HOR    *
1020 REM * ---------------------   *
1030 REM * (X1,Y1)=LINKSONDER      *
1040 REM * (X2,Y2)=RECHTSBOVEN     *
1050 REM * D=AFSTAND STREPEN       *
1060 REM * SCREEN 2 VERPLICHT      *
1070 REM ***************************
1080 REM
1090 LINE (X1,Y1)-(X2,Y2),,B
1100 FOR X=X1 TO X2:FOR Y=Y1 TO Y2 STEP
D
1110 PSET (X,Y)
1120 NEXT Y,X
1130 RETURN
```

# Arceer rechthoek vertikaal

Het arceren van op beeldscherm getekende figuren is heel lastig te programmeren. Het bedenken van een juiste arceerroutine kost de meeste amateurs enorme hoofdbrekens.

Toch kan een figuur betrekkelijk eenvoudig worden gearceerd wanneer we wat slimmigheden toepassen.

De volgende subroutine dient 'gevoed' te worden met vijf variabelen:

| | |
|---|---|
| X1,Y1 | eerste hoofpunt rechthoek |
| X2,Y2 | tweede hoofdpunt rechthoek |
| D | afstand tussen twee arceerstrepen |

Een GOSUB 1000 doet het verdere werk; de gearceerde rechthoek wordt getekend.

De programmalijst:

```
1000 REM ***************************
1010 REM * ARCEER RECHTHOEK VERT *
1020 REM * ----------------------- *
1030 REM * (X1,Y1)=LINKSONDER     *
1040 REM * (X2,Y2)=RECHTSBOVEN    *
1050 REM * D=AFSTAND STREPEN      *
1060 REM * SCREEN 2 VERPLICHT     *
1070 REM ***************************
1080 REM
1090 LINE (X1,Y1)-(X2,Y2),,B
1100 FOR X=X1 TO X2 STEP D:FOR Y=Y1 TO Y
2
1110 PSET (X,Y)
1120 NEXT Y,X
1130 RETURN
```

# Arceer rechthoek 135 graden

Het arceren van op beeldscherm getekende figuren is heel lastig te programmeren. Het bedenken van een juiste arceerroutine kost de meeste amateurs enorme hoofdbrekens.

Toch kan een figuur betrekkelijk eenvoudig worden gearceerd wanneer we wat slimmigheden toepassen.

De volgende subroutine dient 'gevoed' te worden met vijf variabelen:

| | |
|---|---|
| X1,Y1 | eerste hoofdpunt rechthoek |
| X2,Y2 | tweede hoofdpunt rechthoek |
| D | afstand tussen twee arceerstrepen |

Een GOSUB 1000 doet het verdere werk; de gearceerde rechthoek wordt getekend.

De programmalijst:

```
1000 REM ****************************
1010 REM * ARCEER RECHTHOEK \\\      *
1020 REM * ----------------------   *
1030 REM * (X1,Y1)=LINKSONDER       *
1040 REM * (X2,Y2)=RECHTSBOVEN      *
1050 REM * D=AFSTAND STREPEN        *
1060 REM * SCREEN 2 VERPLICHT       *
1070 REM ****************************
1080 REM
1090 LINE (X1,Y1)-(X2,Y2),,B
1100 FOR X=X1 TO X2:FOR Y=Y1+X MOD D TO
Y2 STEP D
1110 PSET (X,Y)
1120 NEXT Y,X
1130 RETURN
```

# Arceer cirkel 45 graden

Het arceren van op beeldscherm getekende figuren is heel lastig te programmeren. Het bedenken van een juiste arceerroutine kost de meeste amateurs enorme hoofdbrekens.

Toch kan een figuur betrekkelijk eenvoudig worden gearceerd wanneer we wat slimmigheden toepassen.

De volgende subroutine dient 'gevoed' te worden met vier variabelen:

| | |
|---|---|
| X1,Y1 | middelpunt cirkel |
| R | straal cirkel |
| D | afstand tussen twee arceerstrepen |

Een GOSUB 1000 doet het verdere werk; de gearceerde cirkel wordt getekend.

De programmalijst:

```
1000 REM ***************************
1010 REM *    ARCEER CIRCEL ///     *
1020 REM *    ----------------      *
1030 REM *  (X1,Y1)=MIDDELPUNT      *
1040 REM *  R=STRAAL                *
1050 REM *  D=AFSTAND STREPEN       *
1060 REM *  SCREEN 2 VERPLICHT      *
1070 REM ***************************
1080 REM
1090 CIRCLE (X1,Y1),R
1100 FOR X=X1-R TO X1+R:FOR Y=Y1+R-X MOD
 D TO Y1-R STEP -D
1110 IF (X-X1)^2+(Y-Y1)^2>R^2 THEN 1130
1120 PSET (X,Y)
1130 NEXT Y,X
1140 RETURN
```

# Arceer cirkel horizontaal

Het arceren van op beeldscherm getekende figuren is heel lastig te programmeren. Het bedenken van een juiste arceerroutine kost de meeste amateurs enorme hoofdbrekens.

Toch kan een figuur betrekkelijk eenvoudig worden gearceerd wanneer we wat slimmigheden toepassen.

De volgende subroutine dient 'gevoed' te worden met vier variabelen:

| | |
|---|---|
| X1,Y1 | middelpunt cirkel |
| R | straal cirkel |
| D | afstand tussen twee arceerstrepen |

Een GOSUB 1000 doet het verdere werk; de gearceerde cirkel wordt getekend.

De programmalijst:

```
1000 REM ***************************
1010 REM *   ARCEER CIRCEL HOR      *
1020 REM *   -----------------      *
1030 REM * (X1,Y1)=MIDDELPUNT       *
1040 REM * R=STRAAL                 *
1050 REM * D=AFSTAND STREPEN        *
1060 REM * SCREEN 2 VERPLICHT       *
1070 REM ***************************
1080 REM
1090 CIRCLE (X1,Y1),R
1100 FOR X=X1-R TO X1+R:FOR Y=Y1-R TO Y1
+R STEP D
1110 IF (X-X1)^2+(Y-Y1)^2>R^2 THEN 1130
1120 PSET (X,Y)
1130 NEXT Y,X
1140 RETURN
```

# Arceer cirkel vertikaal

Het arceren van op beeldscherm getekende figuren is heel lastig te programmeren. Het bedenken van een juiste arceerroutine kost de meeste amateurs enorme hoofdbrekens.

Toch kan een figuur betrekkelijk eenvoudig worden gearceerd wanneer we wat slimmigheden toepassen.

De volgende subroutine dient 'gevoed' te worden met vier variabelen:

| | |
|---|---|
| X1,Y1 | middelpunt cirkel |
| R | straal cirkel |
| D | afstand tussen twee arceerstrepen |

Een GOSUB 1000 doet het verdere werk; de gearceerde cirkel wordt getekend.

De programmalijst:

```
1000 REM **************************
1010 REM *   ARCEER CIRCEL VERT   *
1020 REM *   ------------------   *
1030 REM * (X1,Y1)=MIDDELPUNT     *
1040 REM * R=STRAAL               *
1050 REM * D=AFSTAND STREPEN      *
1060 REM * SCREEN 2 VERPLICHT     *
1070 REM **************************
1080 REM
1090 CIRCLE (X1,Y1),R
1100 FOR X=X1-R TO X1+R STEP D:FOR Y=Y1-
R TO Y1+R
1110 IF (X-X1)^2+(Y-Y1)^2>R^2 THEN 1130
1120 PSET (X,Y)
1130 NEXT Y,X
1140 RETURN
```

# Arceer cirkel 135 graden

Het arceren van op beeldscherm getekende figuren is heel lastig te programmeren. Het bedenken van een juiste arceerroutine kost de meeste amateurs enorme hoofdbrekens.

Toch kan een figuur betrekkelijk eenvoudig worden gearceerd wanneer we wat slimmigheden toepassen.

De volgende subroutine dient 'gevoed' te worden met vier variabelen:

| | |
|---|---|
| X1,Y1 | middelpunt cirkel |
| R | straal cirkel |
| D | afstand tussen twee arceerstrepen |

Een GOSUB 1000 doet het verdere werk; de gearceerde cirkel wordt getekend.

De programmalijst:

```
1000 REM ***************************
1010 REM *    ARCEER CIRCEL \\\     *
1020 REM *    ----------------      *
1030 REM * (X1,Y1)=MIDDELPUNT       *
1040 REM * R=STRAAL                 *
1050 REM * D=AFSTAND STREPEN        *
1060 REM * SCREEN 2 VERPLICHT       *
1070 REM ***************************
1080 REM
1090 CIRCLE (X1,Y1),R
1100 FOR X=X1-R TO X1+R:FOR Y=Y1-R+X MOD
 D TO Y1+R STEP D
1110 IF (X-X1)^2+(Y-Y1)^2>R^2 THEN 1130
1120 PSET (X,Y)
1130 NEXT Y,X
1140 RETURN
```

# Arceer rechthoek 45 graden

Het arceren van op beeldscherm getekende figuren is heel lastig te programmeren. Het bedenken van een juiste arceerroutine kost de meeste amateurs enorme hoofdbrekens.

Toch kan een figuur betrekkelijk eenvoudig worden gearceerd wanneer we wat slimmigheden toepassen.

De volgende subroutine dient 'gevoed' te worden met vijf variabelen:

| | |
|---|---|
| X1,Y1 | eerste hoofdpunt rechthoek |
| X2,Y2 | tweede hoofdpunt rechthoek |
| D | afstand tussen twee arceerstrepen |

Een GOSUB 1000 doet het verdere werk; de gearceerde rechthoek wordt getekend.

De programmalijst:

```
1000 REM **************************
1010 REM * ARCEER RECHTHOEK ///    *
1020 REM * -------------------     *
1030 REM * (X1,Y1)=LINKSONDER      *
1040 REM * (X2,Y2)=RECHTSBOVEN     *
1050 REM * D=AFSTAND STREPEN       *
1060 REM * SCREEN 2 VERPLICHT      *
1070 REM **************************
1080 REM
1090 LINE (X1,Y1)-(X2,Y2),,B
1100 FOR X=X1 TO X2:FOR Y=Y2-X MOD D TO
Y1 STEP -D
1110 PSET (X,Y)
1120 NEXT Y,X
1130 RETURN
```

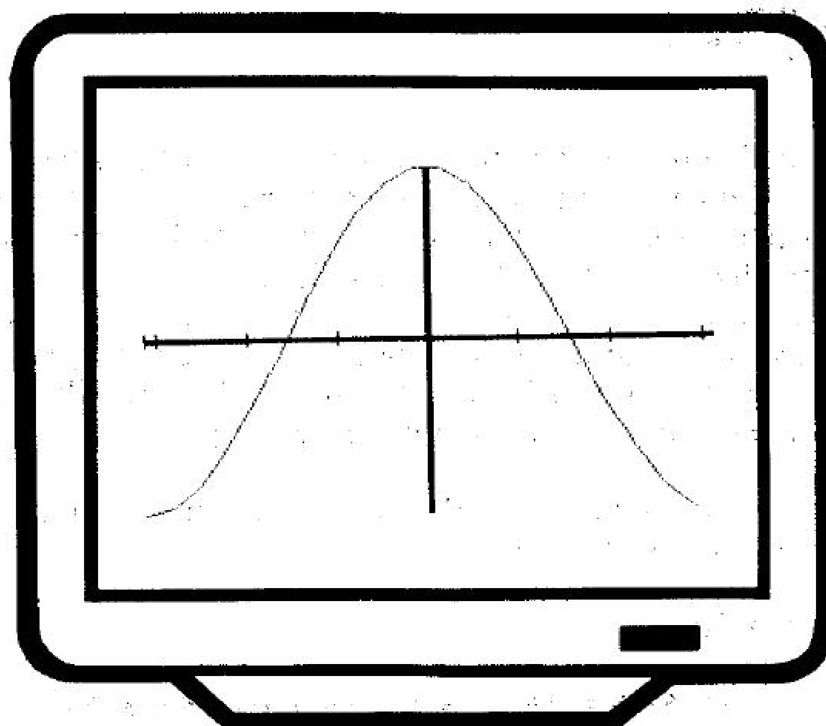
# Lissajous

Een lissajous-figuur is een figuur dat ontstaat op het scherm van een oscilloscoop wanneer men op de horizontale en vertikale afbuigplaten twee wisselspanningen zet. Door deze spanningen in frequentie en fase te laten verschillen, ontstaan de mooiste figuren.
Het volgende programma vraagt u om de beide frequenties en het faseverschil in aantal golflengten. Vervolgens wordt de lissajous-figuur getekend. Door een willekeurige toets in te drukken, wordt het tekenen onderbroken en kunt u opnieuw uw gegevens ingeven.

Voor de leek: intikken en uitproberen. Na enig proberen krijgt u er de meest interessante figuren uit!

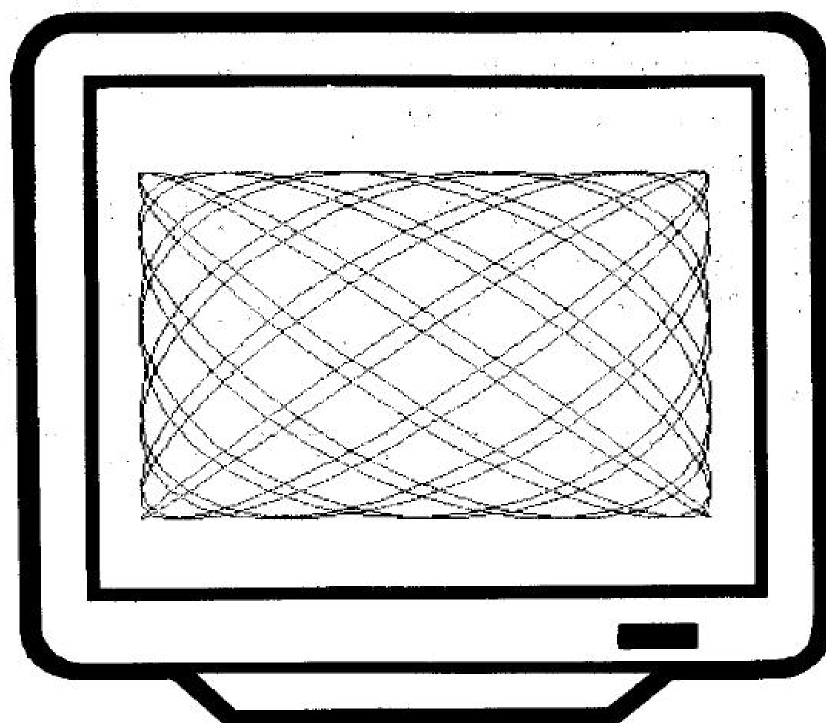
De programmalijst:

```
10 REM ***************************
20 REM *        LISSAJOUS         *
30 REM *        ---------         *
40 REM ***************************
50 REM
60 SCREEN 0:WIDTH 40:KEY OFF:CLS
70 INPUT "FREQUENTIE HORIZONTAAL ";FH
80 INPUT "FREQUENTIE VERTIKAAL   ";FV
90 INPUT "FASEVERSCHIL           ";FA
100 SCREEN 2
110 FOR T=0 TO 1E+20 STEP .4/(FH+FV)
120 X=(COS(T*FH)+1)*128
130 Y=(COS(T*FV+FA*6.283)+1)*96
140 IF M=0 THEN PSET (X,Y):M=1 ELSE LINE
 -(X,Y)
150 IF INKEY$="" THEN NEXT T ELSE RUN
```



Een funktie van de funktieprinter

Een lissajous-tekening



# Funktieprinter

Het volgende programma is een soort 'afkijkertje' voor de middelbare scholier. Dit programma geeft de grafiek van een willekeurige funktie weer.

Op regel 100 dient de laagste X-waarde in I1 te worden geplaatst en op regel 110 dient de hoogste X-waarde in I2 te worden geplaatst. Op regel 120 moet de stapwaarde (de grootte van de stappen waarmee de te bestuderen funktie dient te worden doorlopen) in ST worden gezet. Tenslotte moet op regel 130 de funktie FNF op de juiste wijze worden gedefiniëerd.

Nadat dit allemaal is gedaan, behoeft slechts een RUN te worden ingegeven waarna de computer de funktie tekent. Wanneer mogelijk worden de assen ook getekend. De funktie wordt zo vergroot in X- en Y-richting dat er een duidelijk verloop ontstaat. Op de assen (of, wanneer deze niet getekend kunnen worden, in het midden van het beeld) worden de eenheden afgetekend zodat de vergrotingsfaktor kan worden geschat.

Nadat de natuurlijk eerst uit het hoofd getekende funktie op deze wijze is gecontroleerd, kan een RETURN worden gegeven.
Het programma laat dan nog even de coördinaten zien van de beeldschermhoeken links onder en rechts boven zodat de funktie een plaats kan worden ingeschat ook wanneer de assen niet getekend zijn.

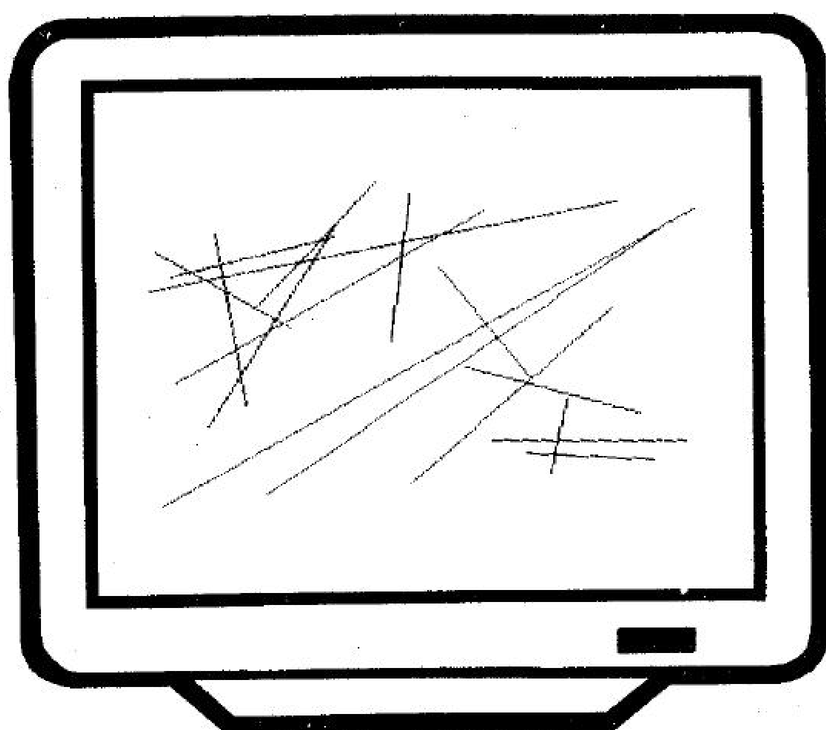
In het programma is een funktievoorbeeld opgenomen dat natuurlijk te allen tijde mag worden veranderd.

De programmalijst:

```
10 REM ****************************
20 REM *    FUNKTIEPRINTER        *
30 REM *    --------------        *
40 REM * I1=VAN WAARDE            *
50 REM * I2=TOT WAARDE            *
60 REM * ST=STAPWAARDE            *
70 REM * FNF=FUNKTIE              *
80 REM ****************************
```

```
90 REM
100 I1=-3.14
110 I2=3.14
120 ST=.3
130 DEF FNF(X)=COS(X)
140 SCREEN 2
150 MI=1E+62:MA=-1E+62
160 FOR X=I1 TO I2 STEP ST
170 Y=FNF(X):IF MI>Y THEN MI=Y
180 IF MA<Y THEN MA=Y
190 NEXT X
200 EX=256/(I2-I1):EY=192/(MA-MI)
210 IF I1<=0 AND I2>=0 THEN LY=-I1*EX:LI
NE (LY-1,0)-(LY+1,191),,BF ELSE LY=128
220 FOR I=INT(MI) TO MA:LINE (LY-3,(I-MI
)*EY)-(LY+3,(I-MI)*EY):NEXT I
230 IF MI<=0 AND MA>=0 THEN LX=-MI*EY:LI
NE (0,193-LX)-(255,191-LX),,BF ELSE LX=9
6
240 FOR I=INT(I1) TO I2:LINE ((I-I1)*EX,
195-LX)-((I-I1)*EX,188-LX):NEXT I
250 M=0:FOR I=I1 TO I2 STEP ST
260 X=(I-I1)*EX:Y=(FNF(I)-MI)*EY
270 IF M THEN LINE -(X,192-Y) ELSE PSET
(X,192-Y):M=-1
280 NEXT I
281 IF INKEY$="" THEN 281
290 SCREEN 0:PRINT "MINIMUM:(";I1;",";MI
;")":PRINT "MAXIMUM:(";I2;",";MA;")"
300 STOP
```

# Truuks met het beeldscherm



Wie zei er dat je op het MSX-beelscherm maar 40 tekens kwijt kan?

Zonder truuks die buiten BASIC liggen zult u in het eerste programma al zien dat het echt mogelijk is om op een grafisch scherm (SCREEN 2) maar liefst 42 karakters naast elkaar te projekteren. En dat met een subroutinetje van DRIE regels...

Dat je met het MSX-beelscherm heel aardige grapjes kunt uithalen, bewijzen de programma's die daarop volgen. Karakters kunnen worden geïnverteerd, onderstreept en aangedikt.

Met een stevige subroutine is het zelfs mogelijk om TIEN teksten over het beeldscherm te laten knipperen.

De ondernemende amateur kan op basis van de hierna volgende programmatuur waarschijnlijk nog veel meer effekten uit zijn MSX-computer halen.

# 42 tekens per regel in SCREEN

De meeste MSX-programmeurs weten wel hoe je een tekst op het grafische beeldscherm kunt printen:

```
100 SCREEN 2:OPEN "GRP:" AS 1
110 PRINT #1,"TEST"
120 CLOSE #1
130 ...
```

Deze methode heeft als nadeel dat de letters nogal ver uiteen worden afgedrukt zodat er maar 32 naast elkaar op een regel passen.

De volgende subroutine projekteert de door u opgegeven tekst op een grafisch scherm (SCREEN 2) en wel zo, dat er 42 tekens naast elkaar passen!

U dient deze subroutine als volgt aan te roepen:

```
100 I$="XXYYA---A"
110 GOSUB 15000
```

I$ dient op XX en YY te zijn gevuld met de horizontale en vertikale karakterpositie waarop moet worden afgedrukt. A---A stelt de af te drukken tekst voor.

Een toepassingsvoorbeeld:

```
100 SCREEN 2:LINE(20,20)-(100,
    100),,B
110 I$="0522DIT IS EEN RECHTHO
    EK":GOSUB 15000
120 ...
```

Bovenstaand voorbeeld tekent eerst een rechthoek en zet dan de tekst "Dit is een rechthoek" eronder.

De programmalijst:

```
10 REM ****************************
```

```
20 REM * PRINT 42 TEKENS ONDER *
30 REM * SCREEN 2. I$="XXYYA-A *
40 REM * XXYY=POSITIE X,Y, A-A *
50 REM * IS TE PRINTEN TEKST   *
60 REM *************************
70 REM
15000 X=VAL(LEFT$(I$,2)):Y=VAL(MID$(I$,3
,2))
15010 OPEN "GRP:" AS 1:FOR I=5 TO LEN(I$
)
15020 PRESET (6*(X+I-5),8*Y):PRINT #1,MI
D$(I$,I,1):NEXT I:CLOSE #1:RETURN
```

# Inverteren van karakters

Het volgende programmaatje inverteert de tekens op beeld (SCREEN 0). De tekens worden omgedraaid zodat de voorgrondkleur de achtergrondkleur en de achtergrondkleur de voorgrondkleur wordt.
Bij de normale kleurinstelling (color 15,4,4) betekent dit dat de letters bij ingave op een lichtgekleurde balk worden geprojekteerd.

Dit programma kan als onderdeel in een ander programma worden opgenomen.

Met een willekeurige screen-opdracht wordt het effect weer ongedaan gemaakt.

De programmalijst:

```
10 REM ******************************
20 REM * INVERTEREN VAN KARAKTERS *
30 REM * ------------------------ *
40 REM * ALLE TEKENS BEHALVE SPA- *
50 REM * TIE WORDEN GEINVERTEERD. *
60 REM ******************************
70 REM
80 SCREEN 0:FOR I=BASE(2)+264 TO BASE (2
)+2048
90 VPOKE I,255-VPEEK(I)
100 NEXT I:STOP
```

# Onderstrepen van karakters

Het volgende programmaatje onderstreept de tekens op beeld (SCREEN 0).

Alle geprojekteerde tekens worden automatisch onderstreept.

Dit programma kan als onderdeel in een ander programma worden opgenomen.

Met een willekeurige SCREEN-opdracht wordt het effect weer ongedaan gemaakt.

De programmalijst:

```
10 REM ******************************
20 REM * ONDERSTREPEN VAN KARAKTERS *
30 REM * -------------------------- *
40 REM * ALLE TEKENS BEHALVE SPATIE *
50 REM * WORDEN ONDERSTREEPT.       *
60 REM ******************************
70 REM
80 SCREEN 0:FOR I=BASE(2)+271 TO BASE (2
)+2048 STEP 8
90 VPOKE I,255-VPEEK(I)
100 NEXT I:STOP
```

# Vette letters

Het volgende programmaatje maakt de tekens op beeld vet (SCREEN 1).

Alle geprojekteerde tekens worden automatisch vet uitgevoerd.

Dit programma kan als onderdeel in een ander programma worden opgenomen.

Met een willekeurige SCREEN-opdracht wordt het effect weer ongedaan gemaakt.

De programmalijst:

```
10 REM **************************
20 REM *     VETTE LETTERS      *
30 REM *     ------------       *
40 REM * DE KARAKTERS WORDEN    *
50 REM * VET GEMAAKT.           *
60 REM **************************
70 REM
80 SCREEN 1:FOR I=BASE(4) TO BASE (4)+20
48
90 VPOKE I,VPEEK(I) OR VPEEK(I)/2
100 NEXT I:STOP
```

# Extra vette letters

Het volgende programmaatje maakt de tekens op beeld extra vet (SCREEN 1).

Alle geprojekteerde tekens worden automatisch extra vet uitgevoerd.

Dit programma kan als onderdeel in een ander programma worden opgenomen.

Met een willekeurige SCREEN-opdracht wordt het effect weer ongedaan gemaakt.

De programmalijst:

```
10 REM ***************************
20 REM *     VETTE LETTERS       *
30 REM *     -------------       *
40 REM * DE KARAKTERS WORDEN     *
50 REM * EXTRA VET GEMAAKT.      *
60 REM ***************************
70 REM
80 SCREEN 1:FOR I=BASE(4) TO BASE (4)+20
48
90 VPOKE I,VPEEK(I) OR VPEEK(I)/2 OR VPE
EK(I)/4
100 NEXT I:STOP
```

# Knipperteksten

Het is in MSX-basic helaas niet mogelijk om delen van een tekst zonder meer te laten knipperen.

Het volgende programma geeft u een mogelijkheid om met een omweg maximaal tien verschillende tekstdelen op het beeldscherm te laten knipperen.

Om een knippertekst aan te zetten, dient u I$ te vullen met de te knipperen tekst, voorafgegaan door vier numerieke posities die de locatie van die tekst op beeld aangeven.

Wilt u de tekst "TRUUKS EN TIPS" bijveoorbeeld op de vierde regel vanaf de vijfde postitie laten knipperen, dan probeert u:

```
XXXX I$="0405TRUUKS EN TIPS":GOSUB 10000
```

Deze tekst zet u weer uit door het volgende te programmeren·

```
XXXX I$="0405":GOSUB 10000
```

Maximaal tien teksten kunnen op deze wijze tegelijk knipperen. Bij het uitvoeren van een (LINE) INPUT of een soortgelijk kommando, houdt het knipperen op. Door de in dit boek opgenomen ingaveroutine te gebruiken, kunt u een tekst tijdens ingave door laten knipperen.

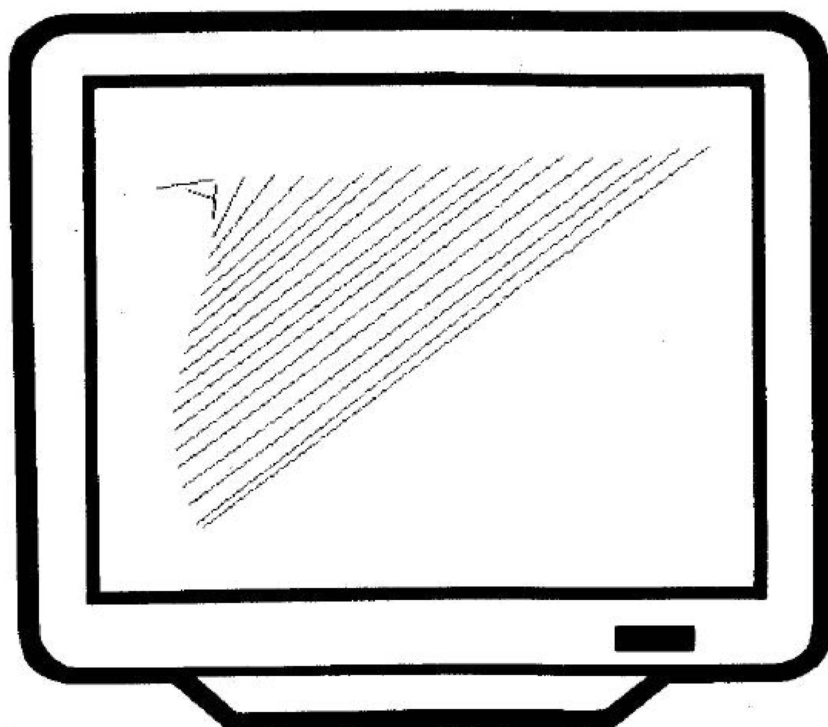
Om deze routine uit te proberen, kunt u het beste het volgende programmeren:

```
100 I$="1010KNIPPERTEST"
110 GOSUB 10000
120 I$="1015GEEF RETURN"
130 GOSUB 10000
140 IF INKEY$="" THEN 140
150 I$="1010":GOSUB 10000
160 I$="1015":GOSUB 10000
170 STOP
```

De programmalijst:

```
10 REM ***************************
20 REM *    10 KNIPPERTEKSTEN      *
30 REM *    -----------------      *
40 REM * I$="XXYYA---A"            *
50 REM *                           *
60 REM * XX/YY=HOR/VERT POSITIE *
70 REM * A---A=KNIPPERTEKST        *
80 REM * GOSUB 10000=KNIPPEREN   *
90 REM * OF WEER UIT ZETTEN        *
100 REM***************************
110 REM
10000 INTERVAL STOP:IF KM=0 THEN KM=-1:O
N INTERVAL=25 GOSUB 10030:DIM KN$(9)
10010 FOR KI=0 TO 9:IF KN$(KI)=I$ THEN L
OCATE VAL(LEFT$(I$,2)),VAL(MID$(I$,3,2))
:PRINT SPACE$(LEN(I$)-4);:KN$(KI)="":INT
ERVAL ON:RETURN ELSE NEXT KI
10020 FOR KI=0 TO 9:IF KN$(KI)="" THEN K
N$(KI)=I$:INTERVAL ON:RETURN ELSE NEXT K
I:INTERVAL ON:RETURN
10030 KJ=NOT(KJ):KX=POS(0):KY=CSRLIN:FOR
 KI=0 TO 9:IF KN$(KI)="" THEN GOTO 10050
10040 LOCATE VAL(LEFT$(KN$(KI),2)),VAL(M
ID$(KN$(KI),3,2)):IF KJ THEN PRINT MID$(
KN$(KI),5); ELSE PRINT SPACE$(LEN(KN$(KI
))-4);
10050 NEXT KI:LOCATE KX,KY:RETURN
```

# Enkele bruikbare gebruikersfunkties



De volgende programma's voorzien in een aantal funkties die binnen MSX-basic in één regel zijn te benoemen.

De funkties voorzien in bepaalde behoeften waarmee elke serieuze programmeur wel een keer wordt gekonfronteerd.

Wanneer we behalve spelletjes ook wat serieuzer programmeerwerk willen gaan aanpakken, dan komen we op een gegeven punt onvermijdelijk tot de noodzaak, bepaalde bedragen verantwoord af te ronden.

Het is altijd gemakkelijk om te kunnen kontroleren of een gedane ingave wel numeriek is.

Ook voorloopnullen en achterspaties dienen op een gegeven moment te kunnen worden toegepast.

Uiteindelijk willen we op een gegeven moment dan ook af van die vervelende STR$-funktie die in elk BASIC weer net iets anders werkt.

# Afronden

Bij het schrijven van mooie, administratieve programma's komen we al snel tot de ontdekking dat we nogal vaak bepaalde numerieke waarden moeten afronden.
Wanneer we bijvoorbeeld met percentages over bedragen gaan werken, komen we al snel met tienden en honderdsten van centen te zitten.

De funktie FNR vraagt van u de af te ronden waarde en het aantal decimalen waarop moet worden afgerond. FNR geeft als resultaat een perfekt afgeronde waarde.

Een toepassingsvoorbeeld:

```
200 INPUT "BEDRAG";B
210 INPUT "BTW-%";P
220 PRINT "BRUTO:";FNR ((1+P/1
    00)*B,2):STOP
RUN
BEDRAG ?125.23
BTW-% ?19
BRUTO: 149.02
Break in 220
Ok
```

De programmalijst:

```
10 REM ***************************
20 REM *        AFRONDEN          *
30 REM *        --------          *
40 REM * FNR(X,Y) GEEFT DE AD-    *
50 REM * MINISTRATIEVE AFRON-     *
60 REM * DING VAN X OP Y NA-      *
70 REM * KOMMAPOSITIES            *
80 REM ***************************
90 REM
100 DEF FNR(X,Y)=SGN(X)*INT(ABS(X)*10^Y+
.5)/10^Y
```

# Controle numeriek

Het INPUT-kommando van het MSX-BASIC geeft een vraagteken, of we willen of niet. Wanneer we geen vraagteken wensen, dan dienen we bijvoorbeeld van de LINE INPUT gebruik te maken (of natuurlijk van de in dit boek opgenomen ingaveroutine).

Dit stelt ons al snel voor een probleem. De LINE INPUT staat alleen stringvariabelen toe. Een numerieke ingave moet via de LINE INPUT eerst in een string worden gezet en daarna, via de VAL-funktie, pas in een numerieke variabele. De controle of een string wel met een juiste numerieke waarde is gevuld, is vrij lastig te programmeren. Toch willen we een numerieke ingave in een goed verzorgd programma graag controleren op een juiste ingave.

De funktie FNN geeft ons gelukkig de mogelijkheid om heel eenvoudig te testen of een string een juiste numerieke waarde bevat. FNN dient 'gevoed' te worden met de te controleren string en geeft als resultaat een −1 (goed) of een 0 (fout).

Een toepassingsvoorbeeld:

```
200 PRINT "WAARDE:";
210 LINE INPUT W$
220 IF FNN(W$)=0 THEN 200
```

Het bovenstaande programmaatje heeft, wanneer vooraf de funktie FNN dient 'gevoed' te worden met de te controleren string en geeft als resultaat een −1 (goed) of een 0 (fout).

De programmalijst:

```
10 REM ***************************
20 REM *   CONTROLE NUMERIEK     *
30 REM *   -----------------     *
40 REM * FNN(X$) GEEFT ALS RE-   *
50 REM * SULTAAT -1 WANNEER X$   *
60 REM * VOLLEDIG NUMERIEK IS    *
70 REM * GEVULD EN ANDERS EEN    *
80 REM * 0                       *
```

```
90 REM **************************
100 REM
110 DEF FNN(X$)=(LEN(STR$(VAL(X$)))+(VAL
(X$)>=0)=LEN(X$))
```

# STR$ zonder spaties

De verschillende BASIC-dialecten die er op deze wereld zoal zijn, hebben allemaal hun eigenaardigheden. Vaak lijken de BASIC's in eerste instantie wel op elkaar maar blijken zij bij nadere bestudering op vele punten van elkaar af te wijken.
De STR$-funktie, die welhaast in elk BASIC-dialect voor handen is, is zo'n funktie die per BASIC-dialect verschillen vertoont.

In het MSX-BASIC geeft de STR$-funktie de tussen haakjes vermelde numerieke waarde in string-vorm weer. Een positieve waarde wordt hierbij voorafgegaan door een spatie terwijl een negatieve waarde door een min-teken wordt voorafgegaan.

Juist in dit opzicht (spaties voor, spaties na) verschillen de BASIC-dialecten nogal. Het ene dialect geeft de string spatieloos weer terwijl het andere een spatie vooraf en misschien ook nog wel een spatie achteraf geeft.

De onderstaande funktie FNV$ vormt een alternatief voor de STR$-funktie en geeft de string zonder voor- of achterspaties weer op de manier zoals een STR$-funktie dat overal zou moeten doen.

De programmalijst:

```
10 REM  ***************************
20 REM  *  STR$ ZONDER SPATIES    *
30 REM  *  -------------------    *
40 REM  * FNV$(X) GEEFT X IN      *
50 REM  * STRINGVORM (ALS STR$)   *
60 REM  * MAAR DAN SPATIELOOS     *
70 REM  * WEER. STR$(12) IS       *
80 REM  * B.V. GELIJK AAN " 12"   *
90 REM  * TERWIJL FNV$(12) GE-    *
100 REM * LIJK IS AAN "12"        *
110 REM ***************************
120 REM
130 DEF FNV$(X)=RIGHT$(STR$(X),LEN(STR$(
X))+(X>=0))
```

# Voorloopnullen

In een goed verzorgd, administratief programma is het vaak nodig dat een numerieke waarde met voorloopnullen in een string wordt gezet. Het maken van een juiste routine hiervoor is voor velen nog een hele klus.
De funktie FNN$ geeft de mogelijkheid om in een handomdraai de juiste kodering in een string te verkrijgen.

Een toepassingsvoorbeeld:

```
200 INPUT "ARTIKELNUMMER:";A
210 A$=FNN$(A,6)
220 PRINT A$:STOP
RUN
ARTIKELNUMMER: ?243
000243
Break in 220
Ok
```

FNN$ verlangt van u de in een string te plaatsen waarde en de lengte van de string. Het resultaat is een string met daarin de waarde met voorloopnullen opgenomen.

De programmalijst:

```
10 REM *************************
20 REM *    VOORLOOPNULLEN      *
30 REM *    --------------      *
40 REM * FNN$(X,Y) GEEFT X MET  *
50 REM * VOORLOOPNULLEN OP      *
60 REM * LENGTE Y IN STRING-    *
70 REM * VORM WEER. FNN$(12,6)  *
80 REM * GEEFT B.V. "000012"    *
90 REM * ALS RESULTAAT          *
100 REM *************************
110 REM
120 DEF FNN$(X,Y)=RIGHT$(STR$(X+10^Y),Y)
```

# Aanvullen achterspaties

Wanneer we een programma een beetje willen verzorgen, komen we vaak tot de noodzaak om een stringvariabele met spaties tot een bepaalde, vaste lengte aan te vullen.

Onderstaande funktie FNL$ dient 'gevoed' te worden met de string en de gewenste lengte.
Het resultaat is een stringwaarde die precies op lengte is.

Een toepassingsvoorbeeld:

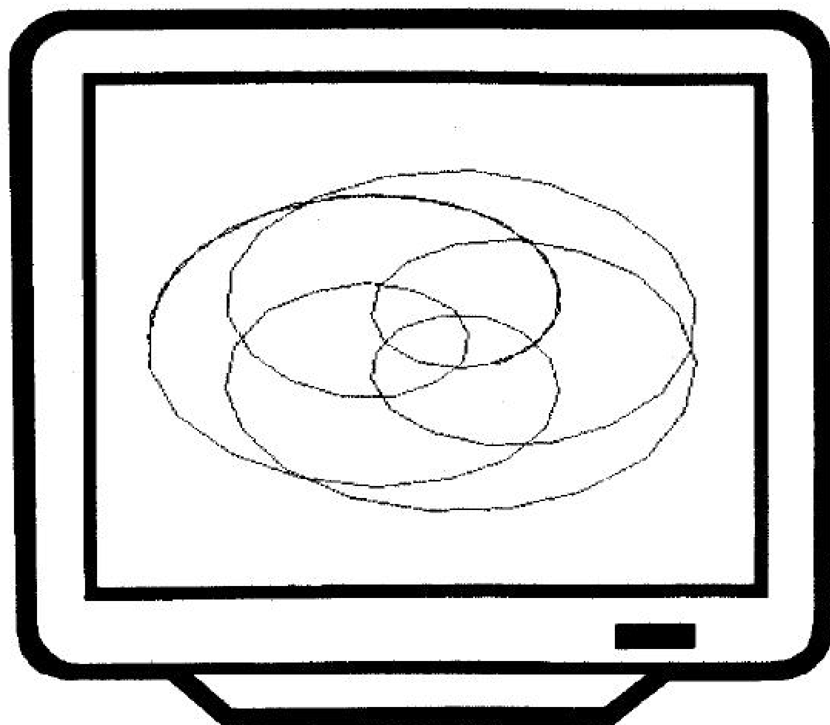
```
100 INPUT "NAAM:";N$
110 N$=FNL$(N$,24)
120 ...
```

In een enkel, klein statement is het met FNL$ nu mogelijk om een string bijvoorbeeld op precies 24 posities af te kappen of aan te vullen.

De programmalijst:

```
10 REM ****************************
20 REM *    STRING OP LENGTE      *
30 REM *    ---------------       *
40 REM * FNL$(X$,X) STELT X$      *
50 REM * MET ACHTERSPATIES OP     *
60 REM * DE LENGTE X              *
70 REM ****************************
80 REM
90 DEF FNL$(X$,X)=LEFT$(X$+SPACE$(X),X)
```

# Datumperikelen



Een datum is een oervervelend ding. Hoe bepaal je of een ingegeven datum wel in orde is? Hoe projekteer je een datum een beetje fatsoenlijk? Hoeveel dagen liggen er precies tussen twee data? Welke dag viel er op een bepaalde datum?

Allemaal vragen die voor de gemiddelde amateur bijna onoverkomelijke problemen vormen en die in de volgende programma's in enkele regels voor u worden opgelost.

De handige programmeur maakt met behulp van de funkties die op de volgende bladzijden staan in een handomdraai een programma dat voor elke willekeurig jaar een komplete kalender op de printer afdrukt.

Wie stuurt ons alvast een kalender voor het jaar 1999?

# Datum formatteren 1

Een datum willen we graag altijd op een speciale wijze .ı beeld brengen. Een datum als 21 januari 1985 noteren we meestal als 21-01-85 of 21/01/85.

Om een datum op deze wijze in MSX-BASIC afgedrukt te krijgen, is nog een hele kunst.

De funktie FNF$ die hieronder is beschreven, formatteert een opgegeven datum voor u op de juiste wijze. U dient de funktie te 'voeden' met een datum en een scheidingsteken en de funktie geeft u de geformatteerde datum als resultaat.

De onderstaande funktie kan bijvoorbeeld met de volgdende programmaregels worden aangevuld:

```
200 INPUT "DATUM";D
210 PRINT "DATUM:";FNF$(D,"-")
220 STOP
RUN
DATUM ?10385
DATUM:01-03-85
Break in 220
OK
```

De programmalijst:

```
10 REM ***************************
20 REM *   DATUM FORMATTEREN     *
30 REM *   -----------------     *
40 REM * FNF$(X,X$) GEEFT DE     *
50 REM * DATUM OP GEBRUIKELIJ-   *
60 REM * KE WIJZE GEFORMAT-      *
70 REM * TEERD. X$ IS HET        *
80 REM * SCHEIDINGSTEKEN.        *
90 REM * FNF$(311285,"-")        *
100 REM * GEEFT STRING           *
```

```
110 REM * STRING "31-12-85" ALS *
120 REM * RESULTAAT               *
130 REM **************************
140 REM
150 DEF FNF$(X,X$)=RIGHT$(STR$(100+INT(X
/10000)),2)+X$+RIGHT$(STR$(100+INT(X/100
)-100*INT(X/10000)),2)+X$+RIGHT$(STR$(10
0+(X-100*INT(X/100))),2)
```

# Datum formatteren 2

De funktie FND$ dient met een datum te worden gevoed. Als uitkomst geeft deze funktie dan de datum in een vorm zoals deze bijvoorbeeld boven een brief behoort te staan.

FND$(101056) geeft bijvoorbeeld als resultaat: "10 oktober 1956"

De programmalijst:

```
10 REM  *************************
20 REM  *  DATUM FORMATTEREN    *
30 REM  *  ----------------     *
40 REM  * FND$(X) GEEFT DE DA-  *
50 REM  * TUM IN WOORDELIJKE    *
60 REM  * VORM.                 *
70 REM  * FND$(010185)          *
80 REM  * OF                    *
90 REM  * 1 JANUARI 1985        *
100 REM *************************
110 REM
120 DEF FND1(X)=INT(X/10000)
130 DEF FND2(X)=INT(X/100)-100*FND1(X)
140 DEF FND3(X)=X-10000*FND1(X)-100*FND2
(X)
150 DEF FND1$(X)=MID$("JANUARI   FEBRUAR
I  MAART     APRIL     MEI       JUNI
   JULI      AUGUSTUS  SEPTEMBER OKTOBER
   NOVEMBER  DECEMBER  ",10*(FND2(X)-1)+
1,10)
160 DEF FND$(X)=STR$(FND1(X))+" "+LEFT$(
FND1$(X),INSTR(FND1$(X)," "))+"19"+RIGHT
$(STR$(FND3(X)+100),2)
```

# Datumcontrole

De funktie FNC geeft de mogelijkheid om een ingegeven datum op juistheid te toetsen. De funktie houdt ook rekening met bijvoorbeeld schrikkeljaren.

Nadat het onderstaande programmaatje is uitgevoerd, kan de anders zo ingewikkelde bepaling of een datum juist is, in een handomdraai worden verricht.

De datum dient in gewone dag-maand-jaarvorm voor funktie FNC te worden opgesteld. Wanneer het resultaat van FNC gelijk is aan −1, dan is de opgegeven datum correct. Is het resultaat echter gelijk aan nul, dan was de opgegeven datum fout.

| | | | |
|---|---|---|---|
| Dus: | FNC(290284) | geeft waarde -1 | (goed) |
| | FNC(290295) | geeft waarde 0 | (fout) |
| | FNC(10182) | geeft waarde -1 | (goed) |

Wanneer de funktie FNC is gedefiniëerd, kan bijvoorbeeld op de volgende manier een datumingave worden gepleegd:

```
300 INPUT "DATUM";D
310 IF FNC(D)=0 THEN 300
```

Bovenstaande regels dragen er zorg voor, dat er net zo lang een datum wordt gevraagd totdat een goede datum is ingegeven.

De programmalijst:

```
10 REM  ****************************
20 REM  *      DATUMCONTROLE       *
30 REM  *      -------------       *
40 REM  * FNC(X) GEEFT EEN 0       *
50 REM  * INDIEN DE DATUM X        *
60 REM  * NIET KLOPT EN EEN -1     *
70 REM  * WANNEER DEZE DATUM       *
80 REM  * WEL KLOPT. X IN DAG-     *
90 REM  * MAAND-JAAR VORM, BIJ-    *
100 REM * VOORBEELD 10184 OF       *
```

```
110 REM * 311285.                  *
120 REM **************************
130 REM
140 DEF FND1(X)=INT(X/10000)
150 DEF FND2(X)=INT(X/100)-100*FND1(X)
160 DEF FND3(X)=X-10000*FND1(X)-100*FND2
(X)
170 DEF FND4(X)=(FND2(X)<13 AND FND2(X)>
0)
180 DEF FND5(X)=30+(FND2(X)-(FND2(X)>7))
 MOD 2 +2*(FND2(X)=2)-(FND3(X) MOD 4=0)
190 DEF FND6(X)=(FND5(X)>=FND1(X) AND FN
D1(X)>0)
200 DEF FNC(X)=-1*FND6(X)*FND4(X)
```

# Dagen tussen twee data

De volgende funktie dient 'gevoed' te worden met twee data. Het resultaat van FNV is het verschil in aantal dagen tussen deze twee data.

Het volgende programmaatje laat zien hoe u FNV bijvoorbeeld kunt gebruiken:

```
200 INPUT "DATUM1";D1
210 INPUT "DATUM2";D2
220 PRINT "DAGEN ERTUSSEN:";FNV(D1,D2)
230 STOP
```

De programmalijst:

```
10 REM ****************************
20 REM * DAGEN TUSSEN TWEE DATA *
30 REM * ---------------------- *
40 REM * FNV(D1,D2) GEEFT HET   *
50 REM * AANTAL DAGEN TUSSEN    *
60 REM * TWEE DATA. D1 EN D2    *
70 REM * ZIJN DE TWEE DATA      *
80 REM * WAARTUSSEN HET VER-    *
90 REM * SCHIL WORDT BEREKEND   *
100 REM ****************************
110 REM
120 DEF FND1(X)=INT(X/10000)
130 DEF FND2(X)=INT(X/100)-100*FND1(X)
140 DEF FND3(X)=X-10000*FND1(X)-100*FND2
(X)
150 DEF FND6(X)=(FND5(X)>=FND1(X) AND FN
D1(X)>0)
160 DEF FNK1(X)=INT(365.25*FND3(X)+FND1(
X)-1+VAL(MID$("0000310590901201511812122
4327330433 4",(FND2(X)-1)*3+1,3))-(FND3(X
) MOD 4 =0)*(FND2(X)<3))
170 DEF FNV(X,Y)=ABS(FNK1(X)-FNK1(Y))
```

# Eeuwigdurende kalender

Dat het MSX-BASIC een verdraaid krachtig BASIC is, bewijst de volgende funktie.

FNK$ is een funktie waarin een eeuwigdurende kalender besloten ligt. De funktie FNK$ dient 'gevoed' te worden met een datum en geeft als resultaat de naam van de dag.

| zo geeft: | het resultaat: |
|---|---|
| FNK$(101056) | woensdag |
| FNK$(010785) | maandag |
| FNK$(300684) | zaterdag |

Nadat funktie FNK$ is gedefiniëerd, kan het volgende programmaatje u bijvoorbeeld precies vertellen op welke dag u bent geboren;

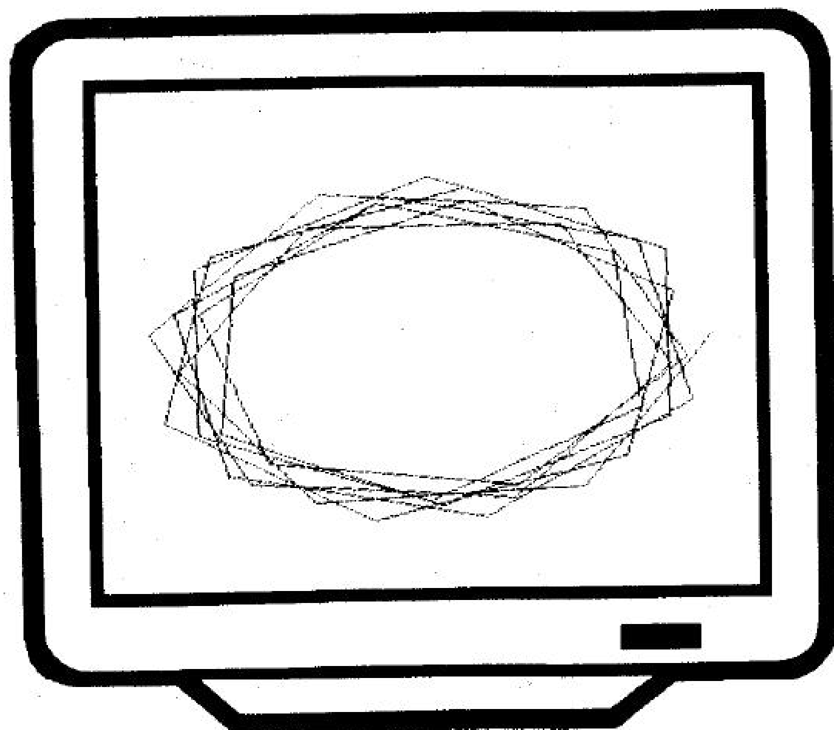
```
200 INPUT "GEBOORTEDATUM";D
210 PRINT "DAG:";FNK$(D)
220 GOTO 200
```

De programmalijst:

```
10 REM ***************************
20 REM * EEUWIGDURENDE KALENDER *
30 REM * ---------------------- *
40 REM * FNK$(X) GEEFT VAN DE    *
50 REM * DATUM IN X DE DAGNAAM   *
60 REM * ZO GEEFT FNK$(101056)   *
70 REM * BIJVOORBEELD HET RE-    *
80 REM * SULTAAT "WOENSDAG "     *
90 REM ***************************
100 REM
110 DEF FND1(X)=INT(X/10000)
120 DEF FND2(X)=INT(X/100)-100*FND1(X)
130 DEF FND3(X)=X-10000*FND1(X)-100*FND2
(X)
```

```
140 DEF FND4(X)=(FND2(X)<13 AND FND2(X)>
0)
150 DEF FND5(X)=30+(FND2(X)-(FND2(X)>7))
 MOD 2 +2*(FND2(X)=2)-(FND3(X) MOD 4=0)
160 DEF FND6(X)=(FND5(X)>=FND1(X) AND FN
D1(X)>0)
170 DEF FNK1(X)=(7+365.25*FND3(X)+FND1(X
)-1+VAL(MID$("000031059090120151181212243
273304334",(FND2(X)-1)*3+1,3))-(FND3(X)
 MOD 4 =0)*(FND2(X)<3)) MOD 7
180 DEF FNK$(X)=MID$("MAANDAG  DINSDAG
WOENSDAG DONDERDAGVRIJDAG  ZATERDAG ZOND
AG    ",FNK1(X)*9+1,9)
```

# Professioneel programmeren



Wanneer u na het doorworstelen van het vorige hoofdstuk nu het programma van uw leven wilt gaan schrijven, dan heeft u zeker wat aan de volgende programma's

De volgende programma-onderdelen werden door u ontworpen om speciaal uw administratieve programmatuur echt professioneel te maken.

Wanneer u de uitgebreide ingave-routine eenmaal hebt ingetikt, kunt u elke ingave in uw programma in één statement helemaal verzorgd regelen. De gebruiker van uw programma (en dat hoeft u natuurlijk niet altijd zelf te zijn) kan niet meer over het beeldscherm heen wandelen met de pijltoetsen. Ook het lelijke 'redo from start' verschijnt nooit meer op uw televisietoestel. Bovendien is het mogelijk om oude gegevens te herzien en om ingaven tot een lengte van 255 tekens zonder een enkel probleem te laten verrichten.

Deze ingaveroutine, de trots van dit boek, is dan ook wel de langste...

Verderop ziet u dat het in MSX-basic mogelijk is om in een handomdraai een kopie van het (alfanumerieke) scherm op de printer te maken.

Het in woorden omzetten van een bedrag (drieëntwintig gulden en vijfennegentig cent) is programmatechnisch een hele toer. Toch blijken er weer niet zoveel programmaregels voor nodig te zijn...

Het vervaardigen van een annuïteitentabel is een werkje dat natuurlijk bij uitstek geschikt is voor de computer.

Tenslotte ziet u aan de hand van een lijst-frame hoe u een deugdelijk lijstprogramma voor uw printer in elkaar moet steken.

# Ingave routine

MSX-basic geeft de gebruiker bij een input-bevel te veel vrijheid. Tijdens een input staat het de gebruiker vrij om bijvoorbeeld met de pijl-toetsen elke richting uit te 'wandelen'.

De volgende routine, die bij uitstek geschikt is voor toepassing binnen wat meer professionele programma's, damt deze vrijheid wat in. De gebruiker kan niet meer zo maar met de pijltoetsen elke kant uit wandelen maar is verplicht om binnen het gestelde aantal karakters te blijven.

Een uniek voordeel van deze routine is dat ingaven tot een lengte van 255 tekens geheel verzorgd kunnen worden uitgevoerd. Wanneer de ingave buiten het beeld dreigt te vallen, reageert deze ingaveroutine plotseling als een soort muurkrant. De tekst wordt zo verschoven dat de laatste ingave altijd te zien is. Zo kan met deze routine een ingave van 32 posities op beeldschermlocatie 30,10 nog gemakkelijk worden uitgevoerd.

De routine wordt als volgt aangestuurd:

Voorafgaand dient I$ met een besturingswaarde te worden gevuld:

```
XXXX LET I$="XXYYTTTA---A"
```

Op de plaats van XX dient de horizontale ingavepositie te worden opgegeven. Op de plaats van YY dient de vertikale ingavepositie te worden opgegeven. Op de plaats van TTT dient het aantal maximaal in te geven tekens te worden ingevuld.

Op de plaats van A---A (mag elke lengte hebben) kan uiteindelijk een tekst worden ingevuld die voor de ingave moet worden geplaatst.

Nadat I$ op de juiste wijze is ingevuld, kan door middel van een:

```
XXXX GOSUB 10000
```

de ingave worden bewerkstelligd.

Eventueel kan een aan te passen (oude) waarde in IO$ worden meege-

geven. De uiteindelijke ingave komt in II$ te staan. Wanneer geen oude waarde behoeft te worden meegegeven, dient IO$ voor de GOSUB 10000 te worden leeggemaakt.

De volgende controletoetsen kunnen tijdens ingave worden gebruikt:

| | |
|---|---|
| pijl omhoog /INS | = teken tussenvoegen |
| pijl omlaag /DEL | = teken verwijderen |
| pijl naar links/rechts | = door tekst 'wandelen' |

Twee voorbeelden:

```
1000 I$="0010006POSTKODE":IO$=""
1010 GOSUB 10000
```

In bovenstaand voorbeeld werd een ingave van een postkode mogelijk gemaakt. Er kunnen niet meer dan 6 tekens worden ingegeven. De ingave wordt voorafgegaan door de tekst 'postkode:' (de dubbele punt komt automatisch) en wordt links in het beeld op regel 10 uitgevoerd.

```
1000 I$="00031280MSCHRIJVING":IO$=W$
1010 GOSUB 10000
```

Op regel 3 werd in dit geval een ingave van 128 tekens mogelijk gemaakt. W$ bevat een eventueel aan te passen waarde en wordt in IO$ meegegeven. In dit geval dient de tekst te gaan 'rollen' omdat er natuurlijk niet zomaar 128 tekens op een beeldschermregel kunnen worden ingegeven.

Deze routine vereist de SCREEN 0 instelling terwijl een WIDTH 40 dient te zijn uitgevoerd.

Uiteraard dient met behulp van een CLEAR-statement een voldoende hoeveelheid string-ruimte te zijn ingericht. Een CLEAR 1024 is in een gemiddeld programma meestal wel genoeg.

De programmalijst:

```
10000 REM ****************************
10010 REM *     INGAVEROUTINE        *
```

```
10020 REM *     -------------          *
10030 REM * I$="XXYYTTTA---A"          *
10040 REM *                            *
10050 REM *      XX=HOR. POSITIE       *
10060 REM *      YY=VERT. POSITIE      *
10070 REM *     TTT=AANTAL TEKENS      *
10080 REM *   A---A=INPUT-TEKST        *
10090 REM *                            *
10100 REM * II$=INGAVERESULTAAT        *
10110 REM * IO$=OUDE WAARDE            *
10120 REM *                            *
10130 REM * TOETSEN:                   *
10140 REM * HOME   =WISSEN INGAVE      *
10150 REM * DEL    =TEKEN WISSEN       *
10160 REM * INS    =TEKEN INVOEGEN     *
10170 REM * RETURN =EINDE INGAVE       *
10180 REM *                            *
10190 REM * PIJLEN:                    *
10200 REM * LINKS  =CURSOR LINKS       *
10210 REM * RECHTS =CURSOR RECHTS      *
10220 REM * OMHOOG =TEKEN INVOEGEN     *
10230 REM * OMLAAG =TEKEN WISSEN       *
10240 REM *                            *
10250 REM ****************************
10260 REM
10270 XX=VAL(LEFT$(I$,2)):YY=VAL(MID$(I$
,3,2)):TT=VAL(MID$(I$,5,3)):II$=IO$:IP=L
EN(II$):IP=IP-(IP=0):II$=II$+SPACE$(TT-L
EN(II$)):PG=0
10280 LOCATE XX,YY: PRINT MID$(I$,8);":"
;LEFT$(II$,46-LEN(I$)-XX);
10290 X1=XX+LEN(I$)+IP-7:IF X1>38 THEN L
OCATE XX+LEN(I$)-6,YY:PRINT MID$(II$,X1-
38,40-POS(0)):X1=39
10300 LOCATE X1,YY,1
10310 KK$=INKEY$:IF KK$="" THEN 10310 EL
```

```
SE KK=ASC(KK$)
10320 IF KK=28 THEN IP=IP+1+(IP=LEN(II$)
):GOTO 10290
10330 IF KK=29 THEN IP=IP-1-(IP=1):GOTO
10290
10340 IF KK=30 OR KK=18 THEN II$=LEFT$(I
I$,IP-1)+" "+MID$(II$,IP,LEN(II$)-IP):PR
INT MID$(II$,IP,40-X1);:GOTO 10290
10350 IF KK=31 OR KK=127 THEN IF IP<TT T
HEN II$=LEFT$(II$,IP-1)+MID$(II$,IP+1)+"
 ":PRINT MID$(II$,IP,40-X1);:GOTO 10290
ELSE 10290
10360 IF KK=11 THEN IO$="":GOTO 10000
10370 IF KK<>13 THEN 10420
10380 QQ=128
10390 IF QQ>TT THEN QQ=QQ/2:GOTO 10390
10400 FOR II=LEN(II$) TO 1 STEP -QQ:IF R
IGHT$(II$,QQ)=SPACE$(QQ) THEN II$=LEFT$(
II$,LEN(II$)-QQ):NEXT II
10410 IF QQ=1 THEN RETURN ELSE QQ=QQ/2:G
OTO 10400
10420 IF KK<32 OR KK>126 THEN BEEP:GOTO
10290
10430 MID$(II$,IP,1)=KK$:PRINT KK$;:IP=I
P+1:IF IP>TT THEN IP=TT
10440 GOTO 10290
```

# Screen 0 copy

Het volgende programma is een kleine subroutine die u bij een bestaand programma kunt opnemen. Met deze subroutine kunt u, mits u onder SCREEN 0 werkt, van uw beeldscherminhoud een kopie op uw printer maken. Een simpele GOSUB 10000 in uw programma is genoeg om deze "hardcopy" te maken.

De programmalijst:

```
10 REM ***************************
20 REM *  HARD COPY SCREEN 0     *
30 REM *  ------------------     *
40 REM ***************************
50 REM
10000 FOR R=0 TO 23:FOR T=0 TO 39
10010 C=VPEEK(BASE(0)+R*40+T)
10020 LPRINT CHR$(C);
10030 NEXT T
10040 LPRINT
10050 NEXT R
10060 RETURN
```

# Screen 1 copy

Het volgende programma is een kleine subroutine die u bij een bestaand programma kunt opnemen. Met deze subroutine kunt u, mits u onder SCREEN 1 werkt, van uw beeldscherminhoud een kopie op uw printer maken. Een simpele GOSUB 10000 in uw programma is genoeg om deze "hardcopy" te maken.

De programmalijst:

```
10 REM *************************
20 REM *  HARD COPY SCREEN 1   *
30 REM *  ------------------   *
40 REM *************************
50 REM
10000 FOR R=0 TO 23:FOR T=0 TO 31
10010 C=VPEEK(BASE(5)+R*32+T)
10020 LPRINT CHR$(C);
10030 NEXT T
10040 LPRINT
10050 NEXT R
10060 RETURN
```

# Bedrag in woorden

In een enkel geval kan het nodig zijn dat een bedrag niet alleen in cijfers maar ook in letters dient te worden geschreven. Dit geldt bijvoorbeeld bij het invullen van bijvoorbeeld een cheque of kwitantie.

Het volgende programma zet, precies zoals het hoort, elk willekeurig bedrag onder de fl. 999.999.999.999,99 in een 'letterbedrag' om. Het bedrag fl. 2345,67 wordt bijvoorbeeld omgezet in TWEE DUIZEND DRIEHONDERDVIJFENVEERTIG GULDEN EN ZEVENENZESTIG CENT.

Het uiteindelijke letterbedrag komt in de variabele B$ te staan. Door kleine wijzigingen aan te brengen, kan dit progamma binnen vele andere programma's worden ingepast.

De programmalijst:

```
10 REM ***************************
20 REM *   BEDRAG IN WOORDEN     *
30 REM *   -----------------     *
40 REM ***************************
50 REM
60 CLEAR 1024:DIM A$(27):FOR I=1 TO 27:R
EAD A$(I):NEXT I
70 DATA "EEN","TWEE","DRIE","VIER","VIJF
","ZES","ZEVEN","ACHT","NEGEN","TIEN","E
LF","TWAALF","DERTIEN","VEERTIEN","VIJFT
IEN","ZESTIEN","ZEVENTIEN","ACHTTIEN","N
EGENTIEN","TWINTIG","DERTIG","VEERTIG","
VIJFTIG","ZESTIG","ZEVENTIG","TACHTIG","
NEGENTIG"
80 INPUT "BEDRAG ";B
90 M=0:B$="":IF B<0 THEN B$="MINUS ":B=-
B
100 IF B=0 THEN B$="NIHIL":GOTO 180
110 A=INT(B/1000000000#):B=B-A*100000000
0!:IF A THEN GOSUB 190:B$=B$+"MILJARD "
```

```
120 A=INT(B/1000000#):B=B-A*1000000!:IF
A THEN GOSUB 190:B$=B$+"MILJOEN "
130 A=INT(B/1000):B=B-A*1000!:IF A THEN
GOSUB 190:B$=B$+"DUIZEND "
140 A=B\1:B=B-A:IF A THEN GOSUB 190
150 IF M THEN B$=B$+"GULDEN":IF B THEN B
$=B$+" EN "
160 A=100*B\1:IF A THEN GOSUB 190:B$=B$+
"CENT"
170 B$=B$+"."
180 PRINT B$:GOTO 80
190 M=1:C=A\100:A=A-100*C:IF C THEN B$=B
$+A$(C)+"HONDERD"
200 IF A=0 THEN 240
210 IF A<20 THEN B$=B$+A$(A):GOTO 240
220 C=A\10:A=A-10*C:IF A THEN B$=B$+A$(A
)+"EN"
230 B$=B$+A$(C+18)
240 B$=B$+" ":RETURN
```

# Annuïteiten

Een annuïteitenlening (hypotheek) is een lening waarbij je elke periode een vast bedrag moet aflossen. In dat bedrag zit steeds een gedeelte rente en een gedeelte schuldaflossing. Deze gedeelten bij elkaar geteld geven altijd een vast bedrag: de annuïteit.

Een annuïteitenlening is veilig maar duur. Veilig omdat u steeds op een vast aflossingsbedrag kunt rekenen. Duur omdat u de eerste tijd erg veel rente betaalt en bijna niets aflost.

Het berekenen van een annuïteit is een hele kunst.

In het volgende programma dient u het te lenen bedrag (kapitaal), een rentepercentage en een aantal periodes in te geven. Pas op dat u de rente per periode opgeeft. Is de rente 10% per jaar en heeft u als periode een half jaar genomen, geef dan 5% rente op!

Aansluitend aan uw ingaven verschijnen de berekende gegevens van de lening alsmede een aflossingstabel over het aantal opgegeven periodes.

Door de PRINT in een LPRINT te veranderen, krijgen we de tabel op papier.

De programmalijst:

```
10 REM ***************************
20 REM *    ANNUITEITENTABEL     *
30 REM *    ----------------     *
40 REM ***************************
50 REM
60 KEY OFF:SCREEN 0:WIDTH 40:CLS:PRINT "
ANNUITEITENBEREKENING":LOCATE 0,2
70 DEFFNR(X)=SGN(X)*INT(ABS(X)*100+.5)/1
00
80 INPUT "KAPITAAL              ";K
90 INPUT "RENTEPERCENTAGE       ";P
100 INPUT  "AANTAL ANNUITEITEN ";A
110 P=P/100:N=FNR(K*P*(P+1)^A/((P+1)^A-1
```

```
))
120 CLS:PRINT USING "KAPITAAL : #######.
##";K
130 PRINT USING "RENTE     : ##.## %";P*1
00
140 PRINT USING "ANNUITEIT: #######.##";
N
150 PRINT "AFLOSSINGSSCHEMA:"
160 PRINT "------------------------------
--------":PRINT"JAAR      RENTE  AFLOSSI
NG SCHULDREST":PRINT "------------------
--------------------"
170 RS=K:FOR I=1 TO A:RE=FNR(P*RS):AF=N-
RE:RS=RS-AF:IF I=A THEN AF=AF+RS:RS=0
180 PRINT USING "###: #######.## #######
.## #######.##";I;RE;AF;RS:NEXT I:STOP
```

# Lijstframe

De struktuur van een programma levert vele amateurs vaak een probleem. De vraag: „Hoe pak ik een dergelijk programma nu aan?" wordt dan ook vaak gehoord.

Wanneer een MSX-er in het gelukkige bezit is van een printer, zal hij hierop onder meer ook graag gegevens lijsten. Dit lijsten moet natuurlijk in een beetje mooie vorm gebeuren, bijvoorbeeld met een vast aantal regels per blad, met boven elk blad een kop en het bladnummer en zó, dat er niet op de scheurrand wordt afgedrukt.

Het onderstaande FRAME is geen programma maar is slechts een opzet van hoe een eenvoudig lijstprogramma er uit dient te zien.

Dit FRAME bevat de struktuur van het programma; het feitelijke programma dient in deze struktuur te worden ingebouwd.

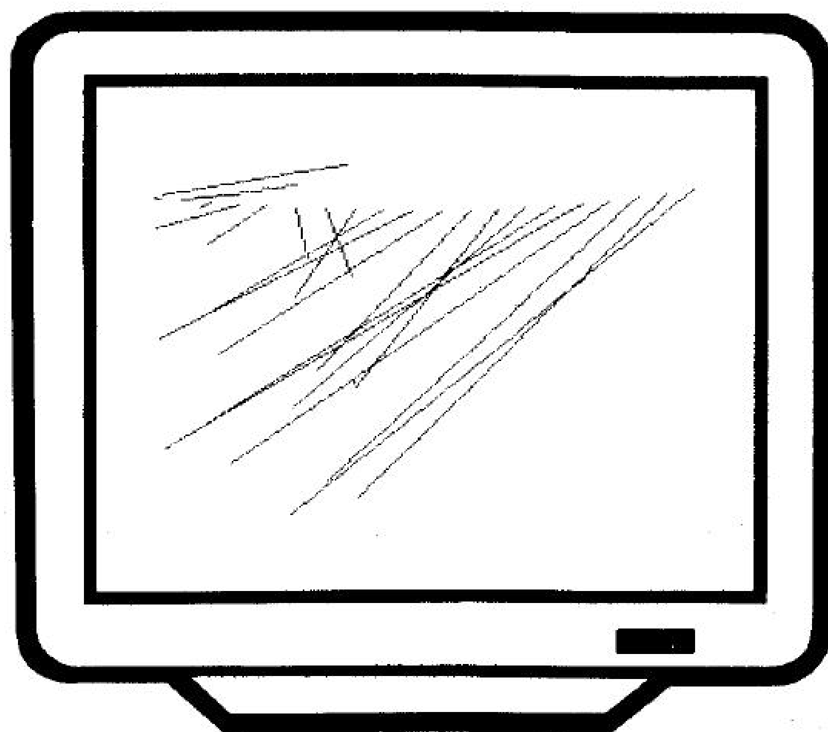
Op regel 70 dient het bestand waaruit de gegevens worden gelijst, te worden geopend. Op regel 80 dient een einde-bestands-controle te worden uitgevoerd. Indien er inderdaad geen gegevens meer af te drukken zijn, dient naar regel 170 te worden gesprongen. Op regel 90-99 dienen alle gegevens te worden ingelezen die nodig zijn voor het afdrukken van een regel. Op regel 110-119 dient het afdrukken van een regel te worden geprogrammeerd. Op regel 120 wordt de regelteller (R) dan met het aantal afgedrukte regels verminderd. Uiteindelijk dient op regel 150-159 de boven de bladzijde te verschijnen kop te worden geprogrammeerd. De variabele B bevat het bladnummer.

De programmalijst:

```
10 REM ****************************
20 REM *         LIJSTFRAME       *
30 REM *         ----------       *
40 REM ****************************
50 REM
60 R=0:B=0
70 REM BESTAND OPENEN
80 REM EERSTE/VOLGENDE GEGEVEN INLEZEN
90 REM EINDE? DAN GOTO 170
```

```
100 IF R=0 THEN R=50:B=B+1:GOSUB 140
110 REM PRINT REGEL
120 REM VERMINDER R MET AANTAL AFGEDRUKT
E REGELS
130 GOTO 80
140 LPRINT CHR$(12)
150 REM KOPTEKST AFDRUKKEN. B=BLADNR.
160 RETURN
170 LPRINT CHR$(12):CLOSE
180 STOP
```

# Karakters op beeld en op de printer



Niet iedere MSX-er heeft een printer. En heeft iemand dan een printer, dan zal het vaak een printer blijken te zijn die niet standaard MSX is.

Zo'n printer heeft vaak een scala van verborgen mogelijkheden.

Met de volgende programma's kunt u onderzoeken wat uw printer nu eigenlijk waard is...

# Karaktertabel

Het volgende programma geeft een overzicht van alle MSX-karakters die op een of andere wijze zijn te verkrijgen.

Door van PRINT een LPRINT te maken, kan eventueel een overzicht op de printer worden verkregen.

Een niet-MSX-printer zal voor vele specifieke karakters echter een totaal ander beeld geven.

De programmalijst:

```
10 REM ***************************
20 REM *       KARAKTERTABEL      *
30 REM *       -------------      *
40 REM ***************************
50 REM
60 SCREEN 0:WIDTH 40:CLS
70 FOR I=0 TO 255 STEP 4
80 PRINT CHR$(1)+CHR$(86);:FOR J=I TO I+
3:PRINT USING " ###: ";J;
90 IF J<32 THEN PRINT CHR$(1)+CHR$(J+64)
; ELSE PRINT CHR$(J);
100 PRINT " ";CHR$(1)+CHR$(86);:NEXT J:P
RINT:NEXT I
```

# Controle karakters 1

Vele MSX-ers hebben aan hun computer een niet-MSX-printer. Dit is geen probleem zolang je maar geen gebruik maakt van de speciale MSX-karakterset.

Zo'n 'vreemde' printer heeft nogal eens mogelijkheden die je op het eerste gezicht niet zou verwachten. Zo kunnen veel printers breed of juist smal afdrukken. Sommige printers kunnen automatisch onderlijnen of naar een andere karakterset overstappen.

Het volgende programmaatje kan u helpen om een gedeelte van deze mogelijkheden op de sporen. Steeds zendt het progamma een zogenaamd controlekarakter naar de printer. Aan u om te zien wat de printer met dit controlekarakter doet.

Wanneer een bepaald effekt u bevalt, noteer dan de op het scherm vermelde CHR$-waarde. Deze waarde kunt u voortaan afdrukken op de printer om het bedoelde effekt te bereiken.

Een heel bekend stuurkarakter is de CHR$(12). In bijna alle gevallen resulteert dit karakter in het opschuiven van het papier over een geheel vel. Door in een programma het kommando:

```
XXXX LPRINT CHR$(12);
```

op te nemen, kan een nieuw vel papier worden worden voorgeschoven.

De programmalijst:

```
10 REM ***************************
20 REM *   TESTPROGRAMMA VOOR     *
30 REM *   CONTROLE KARAKTERS     *
40 REM *   ------------------     *
50 REM ***************************
60 REM
70 SCREEN 0:CLS:WIDTH 40:PRINT"CONTROLEK
ARAKTERS TESTEN OP PRINTER"
```

```
80 FOR I=0 TO 31
90 LOCATE 0,3:PRINT "CONTROLEKARAKTER CH
R$(";MID$(STR$(I),2);") WORDT GETEST."
100 LOCATE 12,6:PRINT "GEEF RETURN"
110 IF INKEY$<>CHR$(13) THEN 110
120 LPRINT CHR$(I);"DIT IS EEN PRINTERTE
ST":NEXT I
```

# Controle karakters 2

Vele MSX-ers hebben aan hun computer een niet MSX-printer. Dit is geen probleem zolang je maar geen gebruik maakt van de speciale MSX-karakterset.

Zo'n "vreemde" printer heeft nogal eens mogelijkheden die je op het eerste gezicht niet zou verwachten. Zo kunnen veel printers breed of juist smal afdrukken. Sommige printers kunnen automatisch onderlijnen of naar een ander karakterset overstappen.

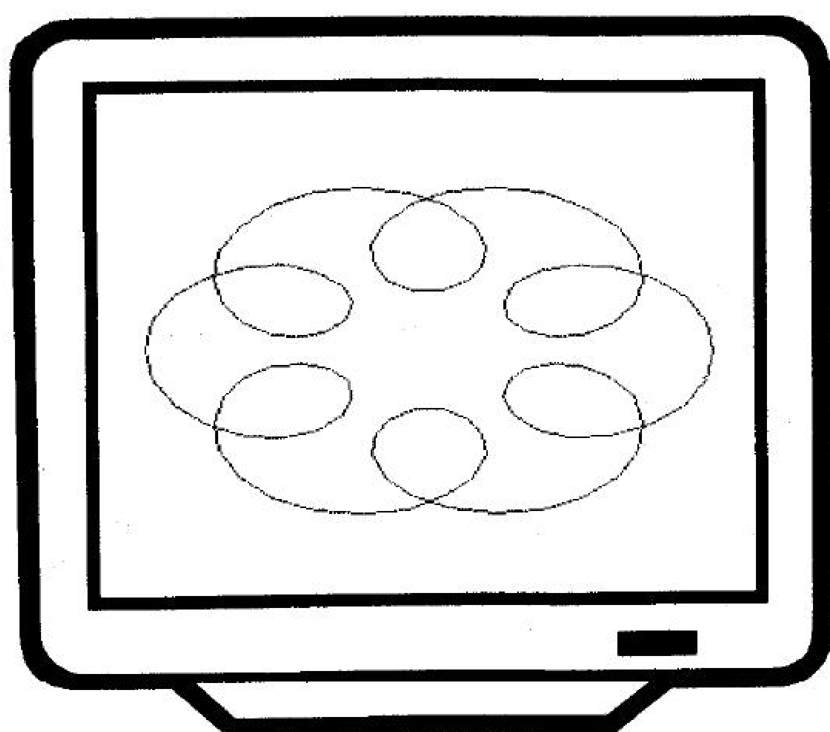
Het volgende programmaatje kan u helpen om een gedeelte van deze mogelijkheden op te sporen. Steeds zendt het programma een zogenaamd escape-karakter naar de printer. Aan u om te zien wat de printer met dit kontrolekarakter doet.

Wanneer een bepaald effekt u bevalt, noteer dan de op het scherm vermelde CHR$-waarden. Deze waarden kunt u voortaan op de getoonde wijze afdrukken op de printer om het bedoelde effekt te bereiken.

De programmalijst:

```
10 REM ***************************
20 REM *  TESTPROGRAMMA VOOR     *
30 REM *   ESCAPE KARAKTERS      *
40 REM *  ------------------     *
50 REM ***************************
60 REM
70 SCREEN 0:CLS:WIDTH 40:PRINT"ESCAPE-KA
RAKTERS TESTEN OP PRINTER"
80 FOR I=0 TO 255
90 LOCATE 0,3:PRINT "ESC-KAR. CHR$(27)+C
HR$(";MID$(STR$(I),2);") WORDT GETEST."
100 LOCATE 12,6:PRINT "GEEF RETURN"
110 IF INKEY$<>CHR$(13) THEN 110
120 LPRINT CHR$(27)+CHR$(I);"DIT IS EEN
PRINTERTEST":NEXT I
```

# Een stapje voorbij BASIC



Behalve in BASIC is een MSX-computer natuurlijk ook in machinetaal te programmeren. Ook kunnen reeds in het ROM aanwezige machinetaalprogramma's met een klein truukje worden aangesproken.

Nee, laat u zich nu niet meteen afschrikken, in de volgende voorbeelden plukt u de vruchten van bepaalde machinetaalprogramma's zonder dat u er veel van hoeft te begrijpen.

We doen een klein stapje buiten het oude vertrouwde MSX-basic en verbazen ons over de mogelijkheden die daar op ons liggen te wachten.

Het doen van een koude start zal niemand ontroeren. Het laten opflitsen van het beelscherm is al veel leuker.

Het via het programma automatisch overstappen van kleine naar grote letters en andersom kan heel goed van pas komen.

Via een klein truukje kunnen we bepaalde processen (muziek, ingave) tot een voortijdig einde brengen.

Grappig is het om te zien dat zelfs het CAPS LOCK lampje vanuit het programma kan worden bestuurd.

Bij lijstprogramma's is het heel handig om vooraf eerst te kunnen bepalen of de printer wel goed staat.

Het kunnen werken met in totaal tien verschillende (alfanumerieke) schermen die in nog geen tiende van een sekonde onderling uitwisselbaar zijn, biedt grote mogelijkheden wanneer men denkt aan het programmeren van bijvoorbeeld een stukje tekstverwerking.

Het ingeven van NEW hoeft niet te betekenen dat u dat programma ook kwijt bent. U kunt het eventueel terughalen...

Al is Microsoft nog zo snel, de amateur achterhaalt haar wel. Een programma waarmee het lekker toch mogelijk is om een ROM-dump op floppy te maken. Moet u wel een floppy eenneid hebben natuurlijk...

Uiteindelijk zijn de fouten die u kunt maken, direkt uit het ROM op te vissen...

# Koude start

Aan het einde van een bepaald programma kan het wel eens wenselijk zijn dat de computer geheel opnieuw opstart, alsof hij uit en aan werd geschakeld.

Zo'n hernieuwde start noemt men in het engels een 'cold boot' ofwel, vrij vertaald, een koude start.

Het volgende, heel simpele routinetje laat zien hoe we koud kunnen starten.

De programmalijst:

```
10 REM *************************
20 REM *     KOUDE START       *
30 REM *     -----------       *
40 REM *************************
50 REM
60 DEF USR=0
70 PRINTUSR(0)
```

# Flitser

Het volgende, hele kleine programma laat uw beeldscherm aan en uit flitsen totdat u een toets ingeeft.

Misschien leuk om in bijvoorbeeld spelprogramma's toe te passen...

De programmalijst:

```
10 REM ***************************
20 REM *         FLITSER         *
30 REM *         --------        *
40 REM ***************************
50 REM
60 DEFUSR0=65:DEFUSR1=68:U=USR(0):FOR I=
1 TO 100:NEXT I:U=USR1(0):FOR I=1 TO 100
:NEXT I:IF INKEY$="" THEN GOTO 60 ELSE S
TOP
```

# Kleine letters/Grote letters

De CAPS-LOCK toets kan op een MSX-computer worden vastgezet om continu hoofdletters te verkrijgen.

Wanneer we later weer kleine letters willen kunnen ingeven, kunnen we deze toets ook weer ontgrendelen.

Deze toets kunnen we via een heel eenvoudig stukje programmatuur ook ontgrendelen en wel als volgt:

De programmalijst:

```
10 REM *****************************
20 REM *      KLEINE LETTERS       *
30 REM *      --------------       *
40 REM *****************************
50 REM
60 POKE 64683!,0
```

**Grote letters**

De CAPS-LOCK toets kan op een MSX-computer worden vastgezet om continu hoofdletters te verkrijgen.

Deze toets kunnen we via een heel eenvoudig stukje programmatuur ook vast zetten en wel als volgt:

De programmalijst:

```
10 REM *****************************
20 REM *      GROTE LETTERS        *
30 REM *      -------------        *
40 REM *****************************
50 REM
60 POKE 64683!,255
```

# Ingavebuffer leeg

In MSX-BASIC kunt u een flink aantal karakters vooruit intoetsen. Deze tekens worden bewaard in een zogenaamde ingavebuffer en gebruikt wanneer er bijvoorbeeld een INPUT-kommando wordt uitgevoerd.

In een enkel geval kan het zo zijn dat vooruitgetoetste gegevens niet te pas komen. Wanneer het programma door een keurig afgevangen fout in een onverwachte ingaveroutine komt, kan vooruit ingetoetste informatie weleens heel vervelend werken.

Het volgende programmaatje toont u hoe u de ingavebuffer in een klap leeg kunt maken.

De programmalijst:

```
10 REM *************************
20 REM *  INGAVEBUFFER LEEG    *
30 REM *  -----------------    *
40 REM *************************
50 REM
60 DEFUSR1=342
70 U=USR1(0)
```

# Muziek uit

Het volgende programma laat u zien hoe u de met PLAY opgestarte muziek voortijdig kunt uitschakelen.

Wanneer u een spelprogramma ontwerpt, dan kan het zijn dat u een bepaald thema wilt laten spelen totdat een toets wordt ingedrukt of een doel wordt geraakt. In dat geval is dit programma nuttig; de uitvoering van het PLAY-kommando wordt onmiddellijk stopgezet.

De programmalijst:

```
10 REM **************************
20 REM *       MUZIEK UIT       *
30 REM *       ----------       *
40 REM **************************
50 REM
60 DEFUSR=144
70 U=USR(0)
```

# Caps-lock knipper

Door het volgende, kleine machinetaal programmaatje kan men het CAPS-LOCK lampje op de MSX-computer aan- of uitzetten. Er wordt echter niet automatisch van hoofdletters op kleine letter omgeschakeld.

In de string R1$ wordt de machinetaalroutine opgeslagen. Deze string mag verder door het programma niet meer worden gebruikt. In verband met automatische geheugenreorganisaties die MSX zo nu en dan uitvoert, is het noodzakelijk om R1$ zo snel mogelijk na een CLEAR-kommando te vullen.

Door in het BASIC-programma na deze routine USR4(0) te gebruiken, wordt automatisch het CAPS-LOCK lampje aangezet. Met een USR4(1) gaat het lampje wwer uit. Wanneer u tussen haakjes een variabele wilt gebruiken, dan dient deze variabele INTEGER te zijn.

Het volgende programma laat het CAPS-LOCK lampje knipperen:

```
200 A%=USR4(A%=0)
210 FOR I=1 TO 100:NEXT I
220 GOTO 200
```

Door de FOR-NEXT loop op 210 te veranderen, kan men het lampje sneller of langzamer laten knipperen.

De programmalijst:

```
10 REM ***************************
20 REM *   CAPS-LOCK KNIPPER     *
30 REM *   -----------------     *
40 REM * USR4(0/1)=AAN/UIT       *
50 REM ***************************
60 REM
70 CLEAR 200:FOR I%=0 TO 32767:READ A%:I
F A%>-1 THEN R1$=R1$+CHR$(A%):NEXT I% EL
SE DEFUSR4=PEEK(VARPTR(R1$)+1)+PEEK(VARP
TR(R1$)+2)*256:DATA 58,248,247,205,50,1,
201,-1
```

# Printer ready routine

Het volgende kleine machinetaal programmaatje maakt het mogelijk om in een BASIC-programma te bepalen of er een printer gebruiksklaar is aangesloten.

Een behulpzame routine wanneer een fool-proof programma dient te worden geschreven.

In de string R3$ wordt de machinetaalroutine opgeslagen. Deze string mag verder door het programma niet meer worden gebruikt. In verband met automatische geheugenreorganisaties die MSX zo nu en dan uitvoert, is het noodzakelijk om R3$ zo snel mogelijk na een CLEAR-kommando te vullen.

Wanneer de funktie USR7(0) als resultaat de waarde 0 geeft, is er geen printer gebruiksklaar aangesloten (er is geen printer of de printer staat niet goed ingesteld). Indien de funktie USR7(0) de waarde 255 geeft, is de printer in orde!

De programmalijst:

```
10 REM ****************************
20 REM * PRINTER READY ROUTINE *
30 REM * ---------------------- *
40 REM * USR7(0) GEEFT 255 BIJ *
50 REM * READY, 0 BIJ NIET RDY *
60 REM ****************************
70 REM
80 CLEAR 200:FOR I%=0 TO 32767:READ A%:I
F A%>-1 THEN R3$=R3$+CHR$(A%):NEXT I% EL
SE DEFUSR7=PEEK(VARPTR(R3$)+1)+PEEK(VARP
TR(R3$)+2)*256:DATA 205,168,0,50,248,247
,201,-1
```

# 9 schermen wisselen

Het volgende machinetaal programma maakt het mogelijk om negen alfanumerieke schermen tegelijk in het geheugen op te slaan. Het opslaan en ophalen van een compleet beeldscherm kost een fraktie van een sekonde.

Dit programma is van bijzonder veel nut wanneer men administratieve programma's wil gaan schrijven (b.v. tekstverwerking).

In de string R2$ wordt de machinetaalroutine opgeslagen. Deze string mag verder door het programma niet meer worden gebruikt. In verband met automatische geheugenreorganisaties die MSX zo nu en dan uitvoert, is het noodzakelijk om R2$ zo snel mogelijk na een CLEAR-kommando te vullen.

Door in een programma de funktie USR6 (beeldschermnummer) op te nemen, wordt het op dat moment zichtbare scherm onder het opgegeven nummer opgeslagen. Een eenmaal opgeslagen beeldscherm kan door gebruik van USR5 (beeldschermnummer) wordt het onder dat nummer opgeslagen beeldscherm weer teruggeplaatst.

Wanneer voor beeldschermnummer een variabele wordt gebruikt, dient deze variabele wel INTEGER te zijn. Een variabele met %-teken (bijvoorbeeld I%) is in ieder geval integer.

Pas op: wanneer een verkeerd beeldschermnummer wordt gebruikt, kan het programma worden opgeblazen!!!

De opgeslagen beeldschermen vereisen 1 kilobyte per stuk. De betreffende geheugenruimte wordt in deze routine met een CLEAR gereserveerd.

De programmalijst:

```
10 REM ****************************
20 REM *  WISSELEN 9 SCHERMEN      *
30 REM *  -----------------------  *
40 REM * OPHALEN = USR5(0/8)       *
50 REM * OPSLAAN = USR6(0/8)       *
```

```
60 REM * 0/8 = SCHERMNUMMER     *
70 REM *************************
80 REM
90 SCREEN 0:CLS:KEY OFF:WIDTH 40:CLEAR 2
00,49151!:DATA 1,0,4,17,0,0,58,248,247,1
35,135,198,192,103,46,0,205,92,0,201,1,0
,4,33,0,0,58,248,247,135,135,198,192,87,
30,0,205,89,0,201,-1
100 FOR I%=0 TO 32767:READ A%:IF A%>-1 T
HEN R2$=R2$+CHR$(A%):NEXT I% ELSE I=VARP
TR(R2$):I=PEEK(I+1)+256*PEEK(I+2):DEF US
R5=I:DEF USR6=I+20:FOR A%=0 TO 8:A%=USR6
(A%):NEXT A%
```

# Old

Enkele BASIC-talen hebben het kommando OLD. Dit kommando haalt een per ongeluk met NEW gewist programma weer terug.

Het per ongeluk intikken van NEW gebeurt iedereen wel een keer en dan meestal net wanneer men nog niet heeft geSAVEd. Een OLD-kommando is op zo'n moment goud waard.

MSX-basic kent helaas geen OLD-kommando. Door een truukje echter is het toch mogelijk om een met NEW gewist programma weer terug te halen. Dit gaat als volgt.

ONMIDDELLIJK na de per ongeluk ingegeven NEW dient het volgende te worden ondernomen:

Tik in:

```
POKE 32770,128
```

```
LIST
```

De eerste regel van het verloren programma verschijnt. Onthou het nummer van deze eerste regel.

Tik nu in:

```
POKE 32769,1
```

en verwijder de eerste regel door alleen het regelnummer in te geven.

Geef vervolgens in:

```
LIST
```

Het programma wordt op beeldscherm gelijst; het eerste deel van de reddingsoperatie is achter de rug. We zijn er echter nog niet; het programma kan nog niet zomaar worden geSAVEd. Save het programma nu niet op de normale wijze maar gebruik de A-optie. Het bijna geredde programma kan bijvoorbeeld als volgd worden geSAVEd:

```
SAVE "TEST",A
```

Nadat het programma op deze wijze is geSAVEd, de computer uit en weer aan zetten en het programma weer laden met het LOAD-kommando. Pas wanneer dat goed lukt, is het programma definitief gered!

Werk wel heel nauwkeurig; een enkele tikfout en het programma is definitief verloren...

# Error-lijst

Het volgende programmaatje haalt de in het standaard MSX mogelijkerwijs optredende fouten direkt uit het ROM-geheugen. Een klein, makkelijk in te tikken programma en toch de volledige foutlijst!

Wanneer u een printer heeft, verandert u de PRINT gewoon in een LPRINT en u krijgt de lijst op papier...

De programmalijst:

```
10 REM ***************************
20 REM *      ERROR-LIJST        *
30 REM *      -----------        *
40 REM ***************************
50 REM
60 F$=""
70 FOR I=15734 TO 32768!
80 IF PEEK(I)=0 THEN F=F+1-24*(F=25):PRI
NT USING "## = &";F;F$:F$="":IF F=59 THE
N STOP ELSE ELSE F$=F$+CHR$(PEEK(I))
90 NEXT I
100 I=I+1:IF PEEK(I)>0 THEN F$=F$+CHR$(P
EEK(I)):GOTO 100 ELSE RETURN
```

# ROM-dump

De doorgewinterde MSX-er met schijveneenheid heeft vast wel eens geprobeerd om de inhoud van het MSX-BASIC ROM op schijf te zetten om het te kunnen bestuderen (bijvoorbeeld met behulp van een RECOURCE-programma). Echter, het geëigende bevel om dat te doen:

```
100 BSAVE "ROM.DMP",0,32767
```

gaat mis; MICROSOFT heeft haar spullen goed beschermd.

Echter, wie het laatst lacht, lacht het best. Met het volgende programma kan toch in no time een 32 kilobyte dump van het MSX-ROM worden gemaakt!

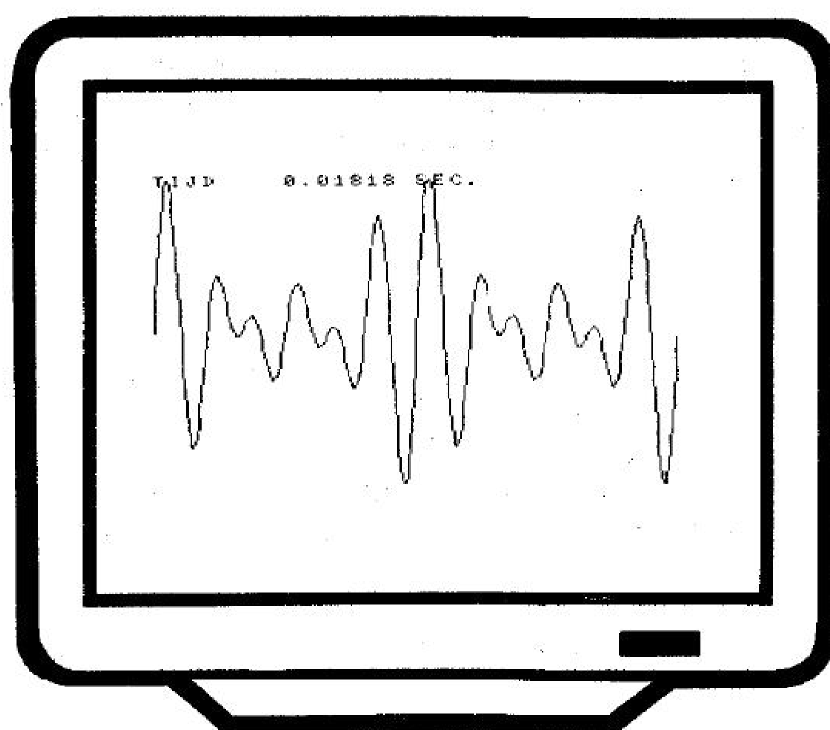
Het geheim schuilt hierin, dat het programma reeds gedeclareerde string-variabelen plotseling in het MSX-ROM laat 'pointen'. Vervolgens worden deze variabelen gewoon op schijf geschreven!

De programmalijst:

```
10 REM ***************************
20 REM *        ROM-DUMP         *
30 REM *        --------         *
40 REM * DUMP DE ROM-INHOUD OP *
50 REM * EEN FLOPPY DISK.        *
60 REM ***************************
70 REM
80 OPEN "ROM.DMP" AS 1
90 FIELD 1,128 AS Q$,128 AS R$
100 A$="":B$=""
110 FOR H=0 TO 127
120 POKE VARPTR(A$),128
130 POKE VARPTR(A$)+1,0
140 POKE VARPTR(A$)+2,H
150 POKE VARPTR(B$),128
160 POKE VARPTR(B$)+1,128
170 POKE VARPTR(B$)+2,H
```

```
180 LSET Q$=A$:LSET R$=B$
190 PUT 1:PRINT ".";
200 NEXT H
210 CLOSE:STOP
```

# Nog wat rekenwerk tot besluit



Het eerste programma stelt u in staat om elke talstelselconversie te maken. Voortaan geen gemier meer met aparte routines voor omrekenen van decimaal naar octaal, van decimaal naar hexadecimaal, van octaal naar binair enzovoorts. Eén allesomvattend programma van nog geen vijftien statements doet het werk voor u...

Het berekenen van priemgetallen is een karweitje dat wel aan uw MSX-computer is toevertrouwd.

Het berekenen van fakulteiten kan in één regel worden geprogrammeerd maar dan moet dit wel recursief gebeuren; een moeilijke techniek die vele hoofdbrekens kost, vooral in BASIC. Velen van u zullen deze recursie, die ook in het talstelselprogramma voorkomt, maar ten dele begrijpen. Gelukkig werken de programma's feilloos en dat is het voornaamste...

In deel 2 van de serie MSX truuks en tips gaan we gewoon verder met nog eens een hondertal bladzijden boordevol nuttige programma's en programmeertips.

# Talstelselconversie

Het volgende programma stelt u in staat om van elk willekeurig talstelsel naar een ander stelsel om te rekenen. Hiertoe vraagt de computer u eerst naar welk ander stelsel u om wilt rekenen. Hierna kunt u getallen opgeven; de computer rekent ze voor u om.

Wanneer u een ingave terug wilt, kunt u de waarde "0" ingeven.

Het laagste talstelsel waarin gewerkt kan worden is (uiteraard) het tweetallig stelsel. Het hoogste talstelsel dat is toegestaan, is het achtendertigtallig stelsel!!!

De programmalijst:

```
10 REM ****************************
20 REM *   TALSTELSELCONVERSIE    *
30 REM *   -------------------    *
40 REM ****************************
50 REM
60 POKE 64683!,255
70 INPUT "VAN STELSEL  ";V:IF V=0 THEN S
TOP ELSE IF V>36 THEN 70
80 INPUT "NAAR STELSEL ";N:IF N=0 THEN 7
0 ELSE IF V=N OR N>36 THEN 70
90 INPUT "GETAL ";G$:IF G$="0" THEN 80
100 G=0:FOR I=1 TO LEN(G$):C$=MID$(G$,I,
1)
110 IF C$<"0" OR (C$>"9" AND C$<"A") OR
C$>"Z" THEN 90
120 C=ASC(C$)-48+7*(C$>"@"):IF C>V-1 THE
N GOTO 90
130 G=G*V+C:NEXT I:I=1:N$="":GOSUB 150
140 PRINT G$;"(";V;")=";N$;"(";N;").":GO
TO 90
150 IF N^I<=G THEN I=I+1:GOSUB 150
160 I=I-1:C=INT(G/N^I):G=G-C*N^I:C=C+48-
7*(C>9):N$=N$+CHR$(C):RETURN
```

# Priemgetallen

Priemgetallen zijn getallen die alleen deelbaar zijn door het getal 1 en door zichzelf. Alle overige delingen geven gebroken resultaten.

Het bepalen van priemgetallen, vooral de wat grotere, is een moeilijk rekenwerk. Een geknipt werkje voor de computer dus.

Het onderstaande programma bepaalt de priemgetallen vanaf 1 en blijft dit net zolang doen totdat u het programma (met CTRL-STOP) onderbreekt.

De programmalijst:

```
10 REM *************************
20 REM *    PRIEMGETALLEN      *
30 REM *    -------------      *
40 REM *************************
50 REM
60 WIDTH 40:CLS
70 PRINT "    1    2    3";
80 FOR I=5 TO 1E+40 STEP 2
90 FOR J=3 TO SQR(I) STEP 2
100 IF I/J=I\J THEN GOTO 120
110 NEXT J:PRINT USING "#####";I;
120 NEXT I
```

# Fakulteit

De fakulteit van een getal is zijn vermenigvuldiging naar 1. Zo is 3 fakulteit (we schrijven "3!") gelijk aan 3x2x1=6.
5! is zo gelijk aan 5x4x3x2x1=120.

Vooral wanneer de getallen groot worden, wordt het berekenen van fakulteiten echt een karweitje voor de computer.

De volgende routine laat zien hoe heel kort het fakulteitsprobleem kan worden opgelost.

Deze routine is recursief. Dat wil zeggen dat hij zichzelf aanroept. In eerste instantie lijkt dat een foutje maar het is heel bewust gedaan! En hij werkt, probeer maar eens.

Hoe deze routine werkt, zal velen een raadsel blijven. Dat hij werkt, staat echter vast!

Een "GOSUB 1000" geeft in de variabele F de fakulteit van N.

De programmalijst:

```
10 REM ****************************
20 REM *         FAKULTEIT        *
30 REM *         ---------        *
40 REM ****************************
50 REM
1000 IF N>0 THEN N=N-1:GOSUB1000:N=N+1:F
=F*N:RETURN ELSE F=1:RETURN
```

# In de serie MSX truuks en tips



In de serie MSX TRUUKS EN TIPS worden korte, krachige oplossingen gepresenteerd voor uw programmeerproblemen. Geen flauwekulletjes maar stukjes programmatuur die u nogdig heeft en die u zo kunt toepassen.

Dat in MSX-basic de meeste problemen bijzonder effektief zijn aan te pakken, bewijst de inhoud van dit boek.

Wij zijn ons er echter van bewust dat er bij u programmeerproblemen leven die in deze serie nog niet werden behandeld.

Wanneer u een of meer van deze problemen heeft en u daarbij vermoedt dat er heel veel mede-amateurs met dezelfde problemen zitten (en dat tweede is natuurlijk erg belangrijk), zet uw programmeerproblemen dan eens op papier en stuur ze naar:

Uitgeverij Stark-Texel
Postbus 302
1794 ZG Oosterend

en wie weet wordt uw probleem in een volgend deel vakkundig opgelost...
Als uw vraag of probleem in de vorm van een programma-oplossing door ons wordt uitgewerkt, krijgt u het komplete boekje toegestuurd met daarbij nog een aardige verrassing.

Tot ziens in deel 2 ...